# Den tidlige kodningsteoris historie

- et undervisningsforløb til gymnasiet

Jankvist, Uffe Thomas

*Publication date:*
2008

*Document Version*
Også kaldet Forlagets PDF

*Citation for published version (APA):*
Jankvist, U. T. (2008). *Den tidlige kodningsteoris historie: - et undervisningsforløb til gymnasiet*. Roskilde Universitet. Tekster fra IMFUFA Nr. 459

IMFUFA tekst

- I, OM OG MED MATEMATIK OG FYSIK

# Den tidlige kodningsteoris historie - et undervisningsforløb til gymnasiet

Uffe Thomas Jankvist
januar 2008

**nr. 459 - 2008**

blank

Roskilde University,
Department of Science, Systems and Models, IMFUFA
P.O. Box 260, DK - 4000 Roskilde
Tel: 4674 2263 Fax: 4674 3020

**Den tidlige kodningsteoris historie**
**– et undervisningsforløb til gymnasiet**

Af: Uffe Thomas Jankvist, januar 2008

IMFUFA tekst nr. 459/ 2008 – 73 sider – ISSN: 0106-6242

For faget matematiks vedkommende stiller den gymnasiale bekendtgørelse af 2007 krav om at der som del af undervisningens supplerede stof behandles elementer af matematikkens historie, eksempelvis gennem matematikhistoriske forløb. Nærværende tekst indeholder undervisningsmateriale designet til et sådant matematikhistorisk forløb.

Et af formålene med materialet har været overfor eleverne at belyse nogle af de sider af matematikkens udvikling som KOM-rapporten (Niss & Jensen, 2002) nævner. Eksempelvis at matematikkens udvikling finder sted i tid og rum, at den udvikles af mennesker og ikke blot opstår ud af ingenting samt at udviklingen drives af såvel interne som eksterne faktorer.

Med kodningsteori tænkes i denne forbindelse på det som undertiden også kendes som kanalkodning, det vil sige studiet af fejlkorrigerende (eller fejlrettende) koder. Med kodningsteoriens tidlige historie er der i denne forbindelse tale om de såkaldte Hamming- og Golay-koder. I materialet gennemgås og opstilles de kodningsteoretiske begreber og forudsætninger som eleverne behøver for at kunne forstå funktionen såvel som konstruktionen af den historisk set først præsenterede fejlkorrigerende kode, den såkaldte binære (7,4)-Hamming-kode. Da (7,4)-koden i denne sammenhæng netop er binær præsenteres eleverne også for binære tal og binær aritmetik. Endvidere er (7,4)-koden, såvel som de øvrige diskuterede fejlkorrigerende koder i materialet, en lineær kode, hvorfor eleverne ligeledes præsenteres for linearitetsbegrebet.

Samtidig med at matematikken for de fejlkorrigerende koder præsenteres fremstilles også historien bag disse samt dele af den allerede etablerede matematik som Hamming baserede sit arbejde med koderne på, eksempelvis det generaliserede afstandsbegreb en metrik. Historien udspiller sig fortrinsvist ved Bell Laboratories, hvor Hamming var ansat samtidig med at Shannon lagde sidste hånd på sin teori for matematisk kommunikation i slutningen af 1940'erne. Hamming var på dette tidspunkt bruger af Bell Labs supercomputere og det var arbejdet med disse maskiner der motiverede ham til at udvikle sine fejlkorrigerende koder. Der er altså tale om et stykke *moderne matematik*, og i KOM-rapportens termer tilmed et hvis udvikling i høj grad er drevet af eksterne faktorer.

For at sikre at eleverne selv går i dybden med de historiske aspekter af den præsenterede kodningsteori afsluttes forløbet med en såkaldt essay-opgave. Ideen er her at eleverne i grupper arbejder med at besvare en række stillede opgaver, at de anvender de (matematikhistoriske) værktøjer som stilles til rådighed herfor og at det hele munder ud i en skriftlig rapport. For at vænne eleverne til aktiviteten med at skrive 'dansk stil' i matematiktimerne er der undervejs i forløbet også indlagt mindre essay-opgaver.

Forløbet om kodningsteoriens tidlige historie er som led i min ph.d. blevet implementeret i en 2g-klasse på Ørestad gymnasium i foråret 2007. Klassens egen matematikunderviser gennemførte forløbet, mens jeg observerede og videofilmede hele processen. Specielt fulgte jeg én gruppe bestående af fem elever i forbindelse med deres arbejde med den afsluttende essay-opgave. Resultaterne af denne undersøgelse vil blive fremlagt i min ph.d.-afhandling som forventes afsluttet i midten af 2009.

Uffe Thomas Jankvist, 2008

# Forord

Dette undervisningsforløb har været kørt i en 2g-klasse som en del af min ph.d. i matematikkens didaktik og matematikkens historie ved Roskilde Universitet. Ph.d.-projektet omhandler brugen af matematikkens historie i matematikundervisningen, hvilket er særlig aktuelt i forbindelse med den ny bekendtgørelse for gymnasiet, hvori der stilles krav om inddragelse af såkaldt 'supplerende stof' i matematikundervisningen. Det hedder således i punkt 2.3:

> For at eleverne kan leve op til de faglige mål, skal det supplerende stof, der udfylder ca. 1/3 af undervisningen, bl.a. omfatte [...] matematik-historiske forløb. (Undervisningsministeriet; 2007)

Det 'faglige mål' som matematikhistoriske forløb retter sig mod lyder:

> Eleverne skal kunne [...] demonstrere viden om matematikkens udvikling i samspil med den historiske, videnskabelige og kulturelle udvikling. (Undervisningsministeriet; 2007)

Det matematikhistorie som I, eleverne, vil blive udsat for i dette forløb er at betegne som *nyere* matematikhistorie, hvilket i denne sammenhæng vil sige matematik hvis udvikling og anvendelse har fundet sted efter 2. verdenskrig.

Undervisningsforløbet handler om den matematik der ligger bag transmission af data. Med transmission af data kan eksempelvis forstås telefonsamtaler fra et sted til et andet, transmission af digitale billeder fra Mars til Jorden eller blot det at sende en SMS fra én mobiltelefon til en anden eller en email fra én computer til en anden. I langt de fleste tilfælde når vi transmitterer data gennemgår disse data en indkodning før de sendes. Overordnet set kan man sige, at der er tre grunde til at indkode data. Den første af disse er *kryptering*, altså at man ved at indkode sine data på en bestemt vis ønsker at hemmeligholde indholdet af disse. Den næste omhandler *effektivitet*, altså at man inden transmissionen påbegyndes komprimerer sine data for på den måde at nedsætte transmissionstiden. Den tredje og sidste grund er *fejlkorrigering*, det vil sige at man gør sig selv i stand til at rette eventuelle fejl der måtte opstå i data under selve transmissionen. Det er denne tredje og sidste grund som dette forløb omhandler.

En essentiel del af dette undervisningsmateriale er den *afsluttende skriftlige opgave* og de såkaldte *essay-opgaver*.

April, 2007
Uffe Thomas Jankvist
IMFUFA, Roskilde Universitet

# Indhold

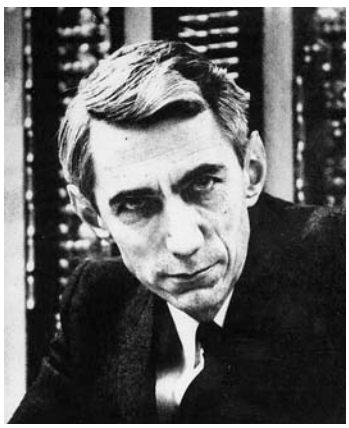
# 1 Introduktion

Teksten i dette undervisningsmateriale er sat med to forskellige skrifttyper; én til matematik, som ser sådan her ud, og den her anvendte til kommentarer, det være sig såvel historiske som anvendelsesorienterede og andre. Dette tiltag er tænkt som en service for læseren.

I en berømt artikel fra 1948 opstillede matematikeren Claude Shannon fra Bell Labs en matematisk teori for kommunikation, en teori der skulle grundlægge de senere matematiske discipliner *informationsteori* og *kodningsteori*. Shannon introducerede det grundlæggende problem, som hans teori skal beskrive på følgende vis:

> Det fundamentale problem i kommunikation består i ét sted at reproducere enten præcist eller tilnærmelsesvist en besked som er valgt et andet sted. I de fleste tilfælde vil beskederne have *mening* [...] Disse semantiske aspekter af kommunikation er irrelevante for det ingeniørmæssige problem. (Shannon; 1948, side 5, oversat fra engelsk)

Fra et matematisk, eller ingeniørmæssigt, synspunkt er det altså ikke indholdet af en besked, det som Shannon omtaler som de semantiske aspekter, der er det essentielle. Vigtigere er det at en besked skal *kommunikeres* fra ét sted til



Claude Elwood Shannon (1916-2001)

Shannon modtog i 1940 sin ph.d.-grad i matematik ved MIT for et arbejde omhandlende populationsdynamik. Inden da havde han studeret såvel matematik som elektroteknik ved University of Michigan. Fra 1941-1972 var Shannon tilknyttet AT&T Bell Telephones i New Jersey som forskningsmatematiker, og det var herfra han udgav sin berømte 1948-artikel. I 1958 blev han udnævnt til Donner Professor of Science ved MIT. Shannon beskæftigede sig også med kunstig intelligens, herunder specielt skakprogrammer, og udgav i 1956 en artikel om den universelle Turingmaskine. Shannon var kendt for at holde sig for sig selv, men blev ofte set kørende rundt på sin ethjulede cykel, undertiden samtidigt jonglerende, til stor fare for sine kollegaer. Shannon skulle efter eget udsagn have arbejdet på en motoriseret kængurustylte, hvilket han hævdede skulle afløse den af kollegaerne frygtede ethjulede cykel. `http://www-groups.dcs.st-and.ac.uk/~history/Mathematicians/Shannon.html`

et andet. At beskrive mængden af information i en besked på matematisk vis er et af de aspekter som informationsteori beskæftiger sig med – blandt andet for at kunne sige noget om, hvor meget en given besked kan komprimeres. Det som vi imidlertid skal koncentrere os om i dette forløb er det som Shannon ovenfor omtaler som »det fundamentale kommunikationsproblem«, nemlig ét sted at reproducere en besked, præcist eller så præcist som muligt, som er afsendt fra et andet sted.

## 1.1 Et kommunikationssystem

Et af Shannons væsentlige bidrag var at han opstillede en *model* for et kommunikationssystem, selv kaldte han det et *skematisk diagram for et generelt kommunikationssystem*[1]. For at forstå indholdet af et sådant skal vi det følgende se på et tænkt eksempel. Men allerførst skal vi have defineret begrebet en *streng*, idet vi ofte skal benytte dette begreb i det følgende. Lad der være givet en mængde $A = \{a_1, a_2, a_3, \ldots, a_n\}$ af symboler, hvor $n$ er et naturligt tal. En sådan mængde kaldes også et *alfabet*. En streng over alfabetet $A$ er da ganske simpelt en endelig følge af elementer fra $A$.

**Eksempel 1.1**
Lad alfabetet være givet ved $A = \{a, b, \alpha, \beta, 1, 2\}$. Eksempler på strenge over $A$ kan da være

$$\beta, \qquad 112, \qquad a\alpha a, \qquad bb\beta 1, \qquad 21\beta\alpha ba, \qquad 22222222222.$$

◇

Lad os begynde med at antage at vi har en *afsender* af et stykke information og at denne afsender er ingen mindre end Zorro. I og med at dette jo er et tænkt eksempel vil vi lade Zorro eksistere i vores moderne verden og forestille os at han ønsker at sende en SMS med informationen $Z$ via sin mobiltelefon til sin største fjende sergent Garcia.

Det første der sker med Zorros meddelelse er en *omsætning til binær information*, det vil sige at et program i mobiltelefonen oversætter det oprindelige bogstav $Z$ til en *binær* streng. Begrebet binær dækker over, at der kun forekommer to forskellige symboler i strengen, eller sagt med andre ord at det alfabet som strengen er taget over kun består af to symboler. Disse symboler er typisk 0 og 1, men kunne i princippet lige så godt være $a$ og $b$ eller ♡ og ♠. Valget af 0 og 1 gør det dog nemmere for os at regne med binære strenge, hvilket vi skal vende tilbage til på et senere tidspunkt. En måde at oversætte bogstaver og tal til strenge af binære symboler på er ved hjælp af det såkaldte standardiserede *ASCII* indkodningsskema fra 1967. ASCII er en forkortelse af »American Standard Code for Information Interchange«. ASCII-skemaet indeholder i alt 128 symboler og et udsnit

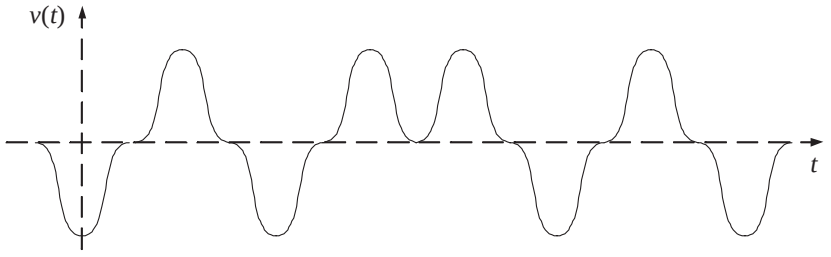
[1] Det oprindelige diagram fra Shannons 1948-artikel kan ses på forsiden af undervisningsmaterialet.

| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| A | → | 01000001 | J | → | 01001010 | S | → | 01010011 |
| B | → | 01000010 | K | → | 01001011 | T | → | 01010100 |
| C | → | 01000011 | L | → | 01001100 | U | → | 01010101 |
| D | → | 01000100 | M | → | 01001101 | V | → | 01010110 |
| E | → | 01000101 | N | → | 01001110 | W | → | 01010111 |
| F | → | 01000110 | O | → | 01001111 | X | → | 01011000 |
| G | → | 01000111 | P | → | 01010000 | Y | → | 01011001 |
| H | → | 01001000 | Q | → | 01010001 | Z | → | 01011010 |
| I | → | 01001001 | R | → | 01010010 | ml.rum | → | 00100000 |

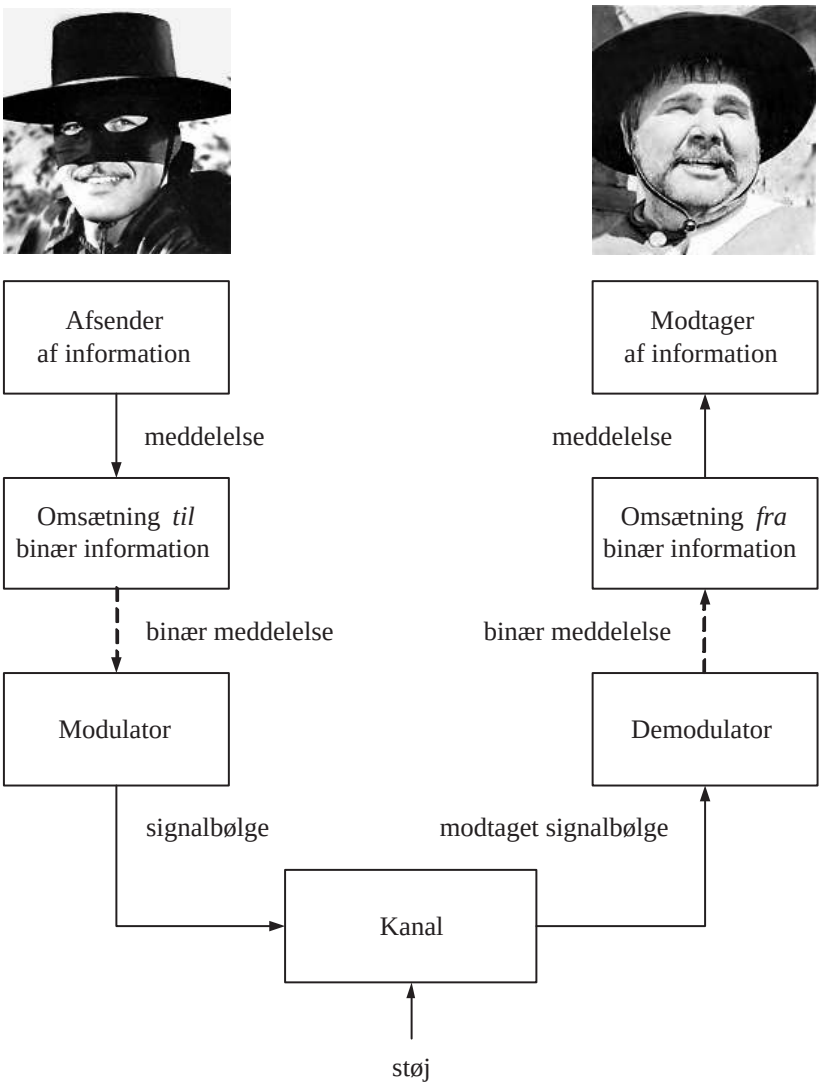
**Tabel 1.1** Et udsnit af ASCII-tegnenes indkodningsskema.

af disse kan ses i tabel 1.1. Ved hjælp af dette skema kan vi oversætte Zorros $Z$ til den binære streng 01011010. De enkelte binære symboler, 0 eller 1, omtales også gerne som *bit*, hvilket er en forkortelse af »binary digit«. En streng af otte bit kaldes en *byte*. Hver repræsentation af et symbol i ASCII fylder altså en byte.

Fordelen ved en binær meddelelse frem for en indeholdende 128 forskellige tegn er, at den er nem at oversætte til en signalbølge, det være sig et elektrisk signal eller en radiobølge, som kan transmitteres gennem et medie, her omtalt som en *kanal*. I vores tilfælde udgør luften (atmosfæren) kanalen, idet Zorros mobiltelefon sender en signalbølge til en satellit som videresender signalbølgen til sergent Garcias mobiltelefon. Måden hvorpå en binær streng oversættes til en signalbølge afhænger af den proces der kaldes *modulation*. Som et simpelt eksempel kan tænkes, at man anvender en enkelt bølgeform til at transmittere de to binære symboler. På figur 1.1 kan ses et eksempel på en sådan bølgeform, nemlig $v(t)$. En bølgetop i $v(t)$, det vil sige et udsving over $t$-aksen, repræsenterer det binære symbol 1 og en bølgedal, det vil sige et udsving under $t$-aksen, repræsenterer det binære symbol 0. En sådan modulation kaldes for *binær modulation*. Den binære modulation af Zorros meddelelse for $Z$, 01011010, kan ses på figur 1.1.




**Figur 1.1** Eksempel på binær modulation af strengen 01011010. En bølgetop repræsenterer 1, mens en bølgedal repræsenterer 0.




**Figur 1.2** Model for kommunikationssystemet med Zorro som afsender og sergent Garcia som modtager. På de to fotografier (1957-59) er det Guy Williams i rollen som Zorro alias Don Diego de la Vega og Henry Calvin i rollen som sergent Demetrio Lopez Garcia.

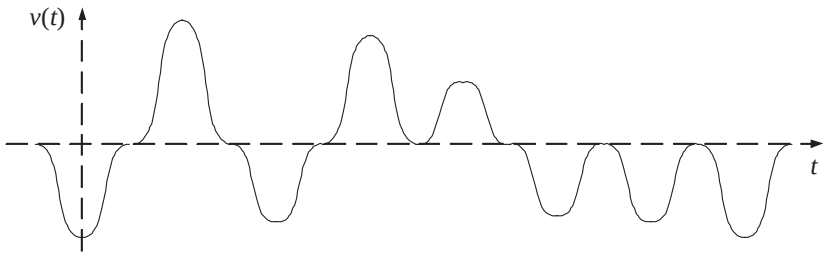
Problemet med signalbølger som sendes gennem en kanal er at de kan blive forstyrret, man siger at de bliver udsat for *støj*. Begrebet støj dækker over alle former for forstyrrelser som kan påvirke signalbølgerne, det være sig atmosfæriske forstyrrelser, radiobølger eller hvad man nu kan forestille sig. Hvis en signalbølge udsættes for støj kan den ændre

udseende, således at den modtagne signalbølge ikke ligner den afsendte. Men lad os se situationen illustreret med en variation af Shannons model for et kommunikationssystem.

Figur 1.2 illustrerer den vej en digital meddelelse må følge fra afsender til modtager. Zorro er altså vores afsender i øverste venstre hjørne og sergent Garcia vores modtager i det øverste højre hjørne. Når sergent Garcias mobiltelefon modtager den afsendte signalbølge bliver denne først *demoduleret*, det til sige oversat fra signalbølge til en binær streng. Derefter foregår en *omsætning fra binær information* til vores almindeligt kendte bogstaver. Dette gøres ved at man i ASCII-skemaet så at sige oversætter tilbage. Når dette er gjort vil sergent Garcia være i stand til at læse SMSen fra Zorro på sin telefons display.

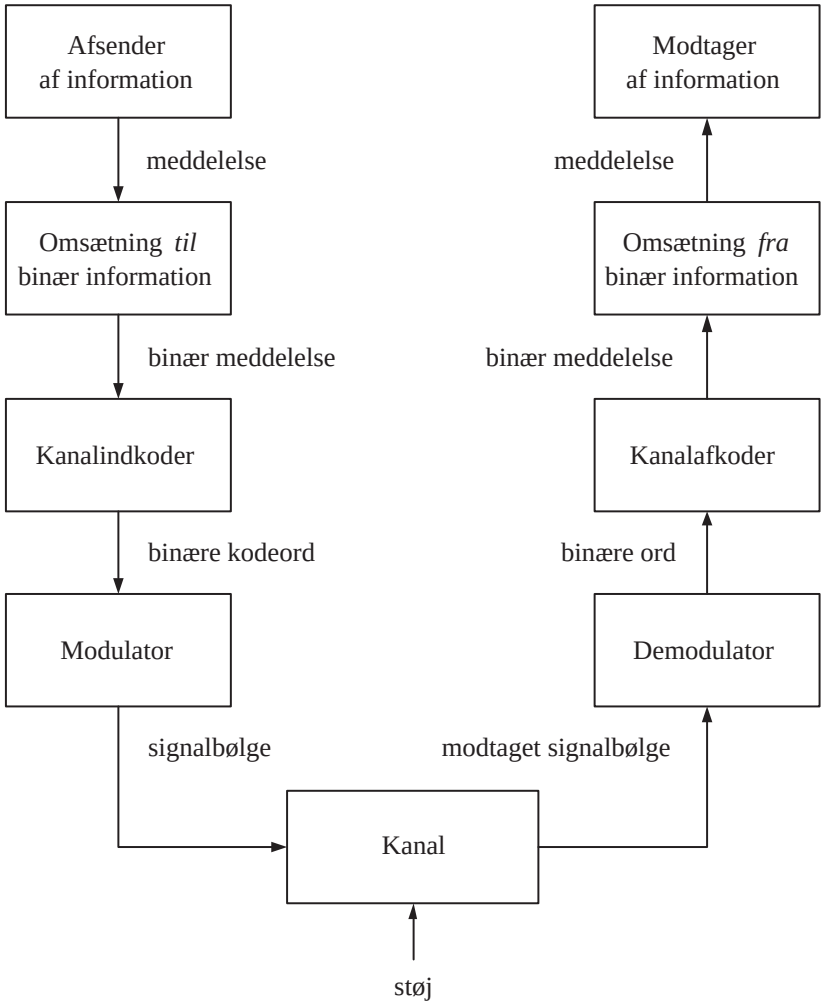
Hvis signalbølgen på figur 1.1 ikke bliver udsat for støj vil sergent Garcia modtage beskeden $Z$, hvorefter han formentlig vil begynde at ryste i bukserne. Hvis signalbølgen derimod bliver udsat for støj vil dette i værste tilfælde give sig til udtryk i, at en bølgetop bliver opfattet af demodulatoren som en bølgedal eller omvendt, hvilket igen kan betyde at den binære meddelelse som demodulatoren spytter ud kan være forskellig fra den afsendte binære meddelelse. Med andre ord er der altså tale om, at et 1-tal kan blive demoduleret som et 0, eller et 0 som et 1-tal.

På figur 1.1 så vi den signalbølge som Zorros mobiltelefon afsendte. Lad os nu sige, at denne bølge udsættes for støj i atmosfæren således at den bølge som sergent Garcias mobiltelefon modtager ser ud som på figur 1.3.




**Figur 1.3** Eksempel på at signalbølgen for den binære streng 01011010 har været udsat for støj.

De bølgetoppe som ikke er blevet forstyrret så meget at de er blevet til bølgedale, det vil sige er røget under $t$-aksen, vil blive demoduleret korrekt. Det tilsvarende gør sig gældende for bølgedale. Signalet på figur 1.3 vil derfor blive demoduleret korrekt på nær for den anden sidste bit. Det vil sige, at den binære meddelelse istedet for at være 01011010 nu bliver 01011000. Denne binære streng bliver ved hjælp af tabel 1.1 omsat fra binær information. Resultatet er ikke $Z$ men derimod $X$, en meddelelse som sergent Garcia står uforstående overfor (med mindre han da tror, at $X$-*Men* er ude efter ham).




**Figur 1.4** Model for et kommunikationssystem som omfatter kanalkodning.

Dette scenario kunne have været undgået, hvis Zorros mobiltelefon havde anvendt *kodningsteori*. (Kodningsteori kendes også under navnet *kanalkodning* og teorien omhandler fortrinsvist bestemmelsen af såkaldte *fejlkorrigerende* eller *fejlrettende* koder). Det helt overordnede princip i kodningsteori er, at man ved transmissionen af den binære meddelelse tilføjer en mængde *ekstra (binær) information*, som gør det muligt at opdage og/eller korrigere fejl i transmissionen. Vi skal i de næste kapitler se, præcist hvordan denne ekstra information kan tilføjes. Tilføjelsen af den ekstra binære information sker ved en proces kaldet *kanalindkodning* og fjernes igen ved en process kaldet *kanalafkodning*. Indkodning og

afkodning sker henholdsvis umiddelbart før modulation og umiddelbart efter demodulation. Altså må vi udvide vores kommunikationssystem fra før til også at omfatte disse to processer. Dette er gjort på figur 1.4. Den kanalindkodede binære meddelelse kaldes for et binært *kodeord*. Demodulatoren i det udvidede system giver os nu ikke længere den binære meddelelse, men derimod et binært *ord*, som måske, måske ikke, er et kodeord, alt afhængig af om signalbølgen er blevet udsat for støj eller ej. Kanalafkoderen korrigerer (så vidt muligt) for fejl i det binære ord ved at oversætte dette til et binært kodeord. Afkoderen fjerner den ekstra binære information fra kodeordet og spytter som resultat en binær meddelelse ud. Tilstedeværelsen af den ekstra binære information gør det altså muligt at opdage og/eller korrigere for eventuelle fejl opstået som følge af støj under transmissionen.

## 1.2 Shannons startskud til kodningsteorien

Fra et kodningsteoretisk synspunkt var det vigtigste bidrag i Shannons 1948-artikel en helt bestemt matematisk sætning, senere undertiden omtalt som *kodningsteoriens hovedsætning*, hvis indhold vi i det følgende skal beskæftige os med.

Når man transmitterer beskeder taler man om at disse har en *hastighed*, hvilket vil sige at der transmitteres et vist antal bit per sekund. Ofte har den kanal som beskeden skal transmitteres igennem en begrænset kapacitet, det vil sige, at den maksimalt kan transmittere et vist antal bit per sekund. Denne kapacitet kendes som *kanal-kapaciteten*. Kodningsteoriens hovedsætning siger da, at hvis man vil transmittere med en bestemt hastighed, og hvis denne hastighed er under kanal-kapaciteten, så findes der et kodningssystem således at sandsynligheden for fejl i den modtagne besked kan gøres vilkårligt lille.

I matematikken kan man undertiden vise eksistensen af noget uden samtidigt at 'pege på' dette. Det vil sige at man kan være i stand til at tale om, at visse ting findes og eventuelt hvordan de opfører sig, uden samtidigt at kunne konstruere disse ting og vise hvordan de ser ud. Shannons resultat er et eksempel på et sådant eksistensudsagn forstået på den måde, at det udelukkende udtaler sig om at der *findes* gode koder, men ikke udtaler sig om hvordan man *konstruerer* disse koder. Og netop det faktum at man vidste at de gode fandtes har nok appelleret til skattejæger-instinktet hos datidens matematikere. I hvert fald begyndte en tiltagende jagt på de gode koder, en jagt der kulminerede med opdagelsen af de såkaldte turbo-koder i 1993, en klasse avancerede og matematisk set endnu ikke helt forståede koder, som man regner med er omtrent så gode som fejlkorrigerende koder overhovedet kan blive. Med hensyn til turbo-koderne og de endnu ikke forståede matematiske aspekter af disse siger den danske kodningsteoretiker Tom Høholdt (DTU):

> De er opfundet af nogle franskmænd som ikke anede noget om kodningsteori, sådan 'nu prøver vi det her' og det virkede. Hvorfor det virkede så godt er der ikke rigtigt nogen der forstår i dag. Man forstår lidt af det – men godt nok til at de kom ind i

> standarden. [...] men sandheden er jo, at normalt hænger den egentlige anvendelse noget bagefter den teoretiske udvikling. Og i tidernes morgen langt bagefter. Men nu går det forholdsvis hurtigt, så tre-fire år efter at turbo-koderne er opfundet, så kommer de ind i standarderne. Det kan man så nogen gange synes er lige hurtigt nok, fordi man ikke forstår dem godt nok, men det er sådan et teori-matematik-synspunkt. Men synspunktet er et andet en gang imellem for hvis det virker, så skal man da bruge det. (Høholdt; 2004)

## 1.3 Eksempler på kodningsteoris anvendelser

Kodningsteori finder ikke kun sin anvendelse inden for mobiltelefoni. Diverse satellitter som kommunikerer med Jorden gør brug af kodningsteori, satellitter til transmission af radiosignaler, TV-signaler, overvågningssatelliter og så videre. Dertil kommer kommunikation med sonder i det ydre rum, robotter (rovere) på Mars for slet ikke at tale om militære kommunikationssystemer. Også computernetværk, med Internettet som det største, gør flittigt brug af kodningsteori.

PC'er, MP3-afspillere og CD-afspillere anvender også kodningsteori. Faktisk forholder det sig sådan, at der digitalt set ikke er forskel på at *sende* information og *gemme* information. Når man gemmer information på sin PC'ers harddisk og på et senere tidspunkt henter det frem svarer den proces som informationen gennemgår nøjagtigt til den på figur 1.4 beskrevne. Når man afspiller en CD er informationerne blot lagret for en i forvejen, men afspilningen består stadig af en afkodning af de på CD'en lagrede, indkodede binære informationer. Støj på kanalen kan i disse tilfælde udgøres af enten fabrikationsfejl på harddisken eller ridser på CD'en. Er der imidlertid for mange ridser på en CD kan det ske at afspilleren ikke er i stand til at afspille den: Mængden af støj på kanalen giver sig til udtryk i et antal af fejl der overstiger antallet, som den fejlkorrigerende kode er i stand til at rette. Denne situation kan gøre sig gældende inden for alle typer af digital kommunikation og vil resultere i enten et *afkodningssvigt*, hvor man ikke er i stand til at afkode og typisk vil modtage en fejlmeddelelse, eller en decideret *afkodningsfejl*, hvor man afkoder til noget forkert, som vi så det i eksemplet med Zorro og sergent Garcia.

Præcis hvor mange fejl en given kode er i stand til at korrigere for, skal vi se nærmere på senere. Men inden da skal vi dvæle lidt ved de binære tal og se, hvordan vi med symbolerne 0 og 1 er i stand til at udtrykke præcis det samme, som vi til hverdag anvender symbolerne 0-9 til.

## 1.4 Binære tal

Det talsystem som vi til hverdag anvender til at repræsentere hele tal er 10-talsystemet. Et tal i dette system er en følge af symboler over alfabetet $\{0, 1, 2, 3, 4, 5, 6, 7, 8, 9\}$, hvor vi dog generelt undlader at anvende 0 som

et begyndende ciffer. Hver position i en sådan følge repræsenterer en potens af tallet 10:

$$\ldots\ 10^4 = 10000,\ 10^3 = 1000,\ 10^2 = 100,\ 10^1 = 10,\ 10^0 = 1.$$

På hver position står der en koefficient, som skal multipliceres på den pågældende potens af 10. Eksempelvis har vi for tallet 1984

$$1 \cdot 10^3,\ 9 \cdot 10^2,\ 8 \cdot 10^1,\ 4 \cdot 10^0,$$

der står for

$$1 \cdot 10^3 + 9 \cdot 10^2 + 8 \cdot 10^1 + 4 \cdot 10^0,$$

idet opskrivningen 1984 underforstår at 10-potenserne skal adderes.

Det binære talsystem fungerer på fuldstændig samme vis, blot med den forskel at et tal i dette system er en følge over det *binære alfabet* $\{0, 1\}$. Hver position i følgen repræsenterer nu en potens af tallet 2:

$$\ldots\ 2^6 = 64,\ 2^5 = 32,\ 2^4 = 16,\ 2^3 = 8,\ 2^2 = 4,\ 2^1 = 2,\ 2^0 = 1.$$

Ønsker vi at opskrive tallet 1984 i det binære talsystem, må vi først finde den største potens af 2 som er mindre end eller lig med 1984. Det er $2^{10} = 1024$ og vi får:

$$1984 = 2^{10} + 960.$$

Den største potens mindre end eller lig 960 er $2^9 = 512$, hvorved vi får:

$$1984 = 2^{10} + 2^9 + 448.$$

$2^8 = 256$ hvorfor $448 = 2^8 + 192$ hvilket igen giver:

$$1984 = 2^{10} + 2^9 + 2^8 + 192.$$

$192 = 2^7 + 64$, hvorfor

$$1984 = 2^{10} + 2^9 + 2^8 + 2^7 + 64.$$

Og da $64 = 2^6$ får vi til sidst:

$$1984 = 2^{10} + 2^9 + 2^8 + 2^7 + 2^6.$$

Hvis vi nummererer positionerne i vores følge fra højre mod venstre,

$$\underbrace{2^{10} = 1024}_{11},\ \underbrace{2^9 = 512}_{10},\ \underbrace{2^8 = 256}_{9},\ \underbrace{2^7 = 128}_{8},\ \underbrace{2^6 = 64}_{7},\ \underbrace{2^5 = 32}_{6},\ \ldots\ \underbrace{2^0 = 1}_{1},$$

har vi da, at der skal stå 1-taller på positionerne 11, 10, 9, 8, 7 og nuller på de resterende positioner. Det binære tal for 1984 bliver da 11111000000. Undertiden skriver man også

$$1984_{10} = 11111000000_2,$$

for at kunne skelne tallene i de forskellige talsystemer fra hinanden. At gå fra det binære talsystem til 10-talsystemet er væsentligt nemmere. Med udgangspunkt i eksemplet ovenfor opskriver og udregner man typisk en sum af 2-potenser:

$$1\cdot 2^{10}+1\cdot 2^9+1\cdot 2^8+1\cdot 2^7+1\cdot 2^6+0\cdot 2^5+0\cdot 2^4+0\cdot 2^3+0\cdot 2^2+0\cdot 2^1+0\cdot 2^0 = 1984.$$

Ligesom i 10-talsystemet kan vi i det binære talsystem også addere, subtrahere, multiplicere og dividere tal. Det eneste vi blot skal holde os for øje i denne sammenhæng er, at i det binære talsystem er $1 + 1 = 10_2$. For en uddybelse af den binære aritmetik se opgaverne.

## 1.5 Binær repræsentation

Binær repræsentation af information er et meget gammelt og velkendt fænomen. Afrikanske stammefolk siges eksempelvis at have sendt information bestående af kombinationer af henholdsvis en dyb og en høj tone. Ligeledes skulle både australske aborigines og stammefolk fra New Guinea tælle i toere[2]. I antikkens Kina, nærmere bestemt i værket *I Ching* fra cirka 1100 før vor tidsregning, anvendes også binær repræsentation af tal omend der ikke er tale om et egentligt talsystem med dertil hørende aritmetik.

Binære tal og aritmetik med disse kan ligeledes spores langt tilbage i tiden. Den indiske matematiker Pingala giver i sit værk *Chhandahshastra*, hvilket betyder noget i retning af 'videnskab om mål' og som regnes for at være fra ca. 300-200 år før vor tidsregning, den første kendte beskrivelse af det binære talsystem. I den vestlige verden er det først med den berømte tyske filosof og matematiker Gottfried Wilhelm von Leibniz (1646-1716) og hans *Explication de l'Arithmétique Binaire* at det moderne binære talsystem bliver endeligt fastlagt. Leibniz interesserede sig også for mekaniske regnemaskiner, i 1974 byggede han en forbedret udgave af Blaise Pascals (1623-1662) mekaniske regnemaskine fra 1643. Leibniz grublede imidlertid videre over, hvordan hans binære aritmetik kunne anvendes i sådanne maskiner, men der skulle gå mange år førend dette blev en realitet.

I 1854 publicerede en anden matematiker ved navn George Boole (1815-1864) sit arbejde *The Laws of Thought*, hvori han opstillede et logisk system der blev kendt som boolesk algebra. Dette system viste sig senere særdeles brugbart i anvendelsen af binær aritmetik i elektroniske kredsløb. En person der indså dette var ingen mindre end Claude Shannon, der som del af sit speciale, *A Symbolic Analysis of Relay and Switching Circuits*, ved MIT (Massachusetts Institute of Technology) i 1938 skitserede de elektroniske kredsløb med relæer og kontakter som var behøvet for at kunne addere og multiplicere binære tal. Dette arbejde skulle senere danne grundlag for designet af additionsmetoder i moderne lommeregnere og computere.

Den formentlig første praktiske anvendelse af binær aritmetik foregik året inden, nærmere bestemt i november 1937, hvor fysikeren George Stibitz

[2] `http://www.is.wayne.edu/olmt/binary/welcome.htm`

ved Bell Labs havde samlet en (relæ-baseret) computer som var i stand til at addere binært, den senere såkaldte *Model I*. I løbet af de kommende år skulle konstruktionen af computere for alvor tage fart. Ikke mindst på grund af 2. verdenskrig og det dermed øgede behov for beregninger i forbindelse med udviklingen af kraftigere våben (atom- og hydrogenbomben), bedre transportmidler og ikke mindst brydning af fjendens krypterede koder (eksempelvis den tyske Enigma-kode). Den person der efter krigen havde den formentlig største betydning for computeres udvikling og endelige design var det ungarsk-amerikanske matematiske geni John von Neumann (1903-1957). Det var von Neumann der indså, at det program som computeren skulle køre kunne gemmes på computeren ligesom de data programmet skulle køres på. (Tidligere havde program og data ikke været gemt i samme lager, programmer var ofte gemt på såkaldte hulkort.) Samtlige computere i dag indeholder en såkaldt von Neumann maskine, hvilket groft sagt er den overordnede struktur som computeren fungerer efter. von Neumann var heller ikke sen til at indse fordelen ved at benytte binær aritmetik i sit arbejde med computere, men gjorde samtidig en del ud af at udvikle decimal-til-binær og binær-til-decimal oversættere, således at operatøren var i stand til at fodre computeren med decimaltal og ligeledes modtage sådanne som resultat.

Som vi skal se i næste kapitel opstod der med introduktionen af computerne netop et behov for fejlkorrigerende koder.

## 1.6 Opgaver

**Opgave 1**
Oversæt følgende ord til binære strenge ved hjælp af ASCII-indkodningsskemaet: SHANNON, BELL LABS, KODNING.

**Opgave 2**
Ved hjælp af ASCII-skemaet ønskes følgende binære streng oversat tilbage til almindelige bogstaver:

$$01000001010100110100001101001001010010 01.$$

Vink: Begynd med at bryde strengen op i blokke med otte symboler i hver.

**Opgave 3**
Find det fulde ASCII indkodningsskema (for eksempel ved at søge på 'ASCII' på nettet). Oversæt da 'Bell Labs' igen og tag denne gang højde for såvel store som små bogstaver.

**Opgave 4**
Oversæt ved hjælp af ASCII-indkodningskemaet beskeden 'OK' til binær information og tegn dernæst signalbølgen for denne.

**Opgave 5**
Antag, at signalbølgen for 'OK' afsendes. På hvilke toppe eller dale skal den afsendte besked have været udsat for støj for, at den modtagne besked er enten 'OO' eller 'KK'.

**Opgave 6**
Antag, at vi sender en signalbølge afsted og at den modtages og afkodes korrekt, hvilket antyder at bølgen ikke har været udsat for støj. Imidlertid ved vi (af en eller anden årsag), at bølgen har været udsat for støj i flere omgange. Giv mulige bud på hvordan det alligevel kan være, at vi har modtaget en bølge som kunne afkodes korrekt. Begrund dine svar.

**Opgave 7**
Find en beskrivelse af Morse-koden (evt. på nettet) og skriv ved hjælp af Morse-koden beskeden 'Samuel Morse'. Forklar hvorfor Morse-koden teknisk set ikke er binær?

**Opgave 8**
Hvad er ifølge Shannon »det fundamentale kommunikationsproblem«?

**Opgave 9**
Forklar med ord hvad der inden for kodningsteori forstås ved en *kanal*. Giv forskellige eksempler.

**Opgave 10**
Forklar med ord begreberne *kanalindkodning* og *kanalafkodning*.

**Opgave 11**
Forklar med ord begreberne *modulator* og *demodulator*.

**Opgave 12**
Forklar (og evt. diskuter) hvad der forstås ved et *kommunikationssystem* og en *model* for et sådant.

**Opgave 13**
En anden form for matematisk kodning omhandler *komprimering*. Komprimering går overordnet set ud på at få en given mængde data til at fylde mindre, eksempelvis ved at fjerne redundante data eller i tilfældet med digitale billeder ved at slette 'overflødige' data. Antag, at vi i figur 1.4 udover kanalkodning også ønsker at udføre komprimering, hvor skal denne proces da indtegnes i figuren? Indtegn processen i figuren.

**Opgave 14**
En tredje form for matematisk kodning omhandler *kryptering*, hvilket går ud på at kode en given besked, således at denne kan holdes hemmelig. Antag, at vi i figur 1.4 udover kanalkodning *og* komprimering også ønsker at foretage kryptering, hvor skal denne proces da udføres i forhold til de to andre processer? Indtegn processen i din modificering af figur 1.4 fra forrige opgave.

**Opgave 15**
Oversæt de binære strenge for A, B og C i ASCII-skemaet til tal i titalsystemet. Giv på baggrund af disse udregninger et bud på de øvrige ti-talværdier af de binære strenge i ASCII-skemaet.

**Opgave 16**
Oversæt følgende tal i ti-talsystemet til tal i to-talsystemet: 2007, 2048, 2049, 4096, 4097, 5000.

**Opgave 17 (Binær addition)**
Lige såvel som man kan addere tal i ti-talsystemet kan man også addere tal i to-talsystemet. Og lige som i ti-talsystemet opererer man i to-talsystemet med en mente, der gælder følgende:

- $0 + 0 = 0$ med mente 0,
- $0 + 1 = 1 + 0 = 1$ med mente 0,
- $1 + 1 = 0$ med mente 1.

Lad os se på et par eksempler.

$$\begin{array}{cccccc} & & \scriptstyle 1 & \scriptstyle 1 & & \\ & & 1 & 0 & 1 & 0 \\ + & & & 1 & 1 & 0 \\ \hline & 1 & 0 & 0 & 0 & 0 \end{array} \qquad \begin{array}{cccccc} & & \scriptstyle 1 & \scriptstyle 1 & & \\ & & & 1 & 1 & 0 \\ + & & & 1 & 1 & 1 \\ \hline & & 1 & 1 & 0 & 1 \end{array}$$

Bemærk, at vi i andet tilfælde støder på $1 + 1 + 1 = 11$, hvilket giver 1 og 1 i mente. Udregn nu følgende:

a. $100 + 101100$.
b. $1111 + 10000$.
c. $1010101 + 10101010$.
d. $10000001 + 1111110$.
e. $1111111111 + 1$.

På lignende vis kan vi også trække binære tal fra hinanden. Udregn følgende:

f. $111 - 110$.
g. $1010 - 1110$.

**Opgave 18 (Binær multiplikation)**
Vi kan selvfølgelig også multiplicere i to-talsystemet. Her gælder ikke overraskende følgende:

- $0 \cdot 0 = 0$,
- $0 \cdot 1 = 1 \cdot 0 = 0$,
- $1 \cdot 1 = 1$.

Lad os se et par eksempler:

$$\begin{array}{cccccccc} 1 & 0 & 1 & 0 & \cdot & 1 & 1 & 0 \\ \hline & & & & & & & \\ & & & & 0 & 0 & 0 & 0 \\ & & & 1 & 0 & 1 & 0 & \\ + & & 1 & 0 & 1 & 0 & & \\ \hline & & 1 & 1 & 1 & 1 & 0 & 0 \end{array} \qquad \begin{array}{cccccccc} 1 & 1 & 0 & \cdot & 1 & 1 & 1 & \\ \hline & & \scriptstyle 1 & \scriptstyle 1 & & & & \\ & & & & 1 & 1 & 0 & \\ & & & 1 & 1 & 0 & & \\ + & & 1 & 1 & 0 & & & \\ \hline & 1 & 0 & 1 & 0 & 1 & 0 & \end{array}$$

Udregn nu følgende:

a. $100 \cdot 101100$.

b. $1111 \cdot 10000$.

c. $1010101 \cdot 10101010$.

d. $10000001 \cdot 1111110$.

e. $1111111111 \cdot 11$.

På lignende vis kan vi også dividere binære tal. Udregn følgende:

f. $111100 : 110$.

g. $101010 : 111$.

**Opgave 19 (Essay-opgave)**

I denne opgave er det meningen, at I skal skrive et par sider (på computer!) om binære tal, binær repræsentation og disse emners historie. I kan tænke på denne opgave som en form for 'dansk stil', hvor emnet blot er af matematikhistorisk art – lidt a la afsnit 1.5 i dette kapitel.

Ideen er, at I bruger Internettet (eller måske det lokale bibliotek) til at skaffe jer information, for eksempel ved at søge på 'history of binary numbers', 'history of binary representation' eller lignende søgeord. I kan selvfølgelig bruge al den viden I indtil videre har tilegnet jer fra dette undervisningsmateriale.

Med udgangspunkt i de fremskaffede informationer ønskes det, at I forsøger at besvare følgende spørgsmål:

a. Hvornår, historisk set, fremkom ideerne til binære tal og binær repræsentation?

b. Hvorfra, eller fra hvem, kom ideerne til binære tal og binær repræsentation?

c. Hvorfor fremkom ideerne til binær repræsentation?

d. Hvorfor anvender man i dag såvel som i tidligere tider binære tal og binær repræsentation?

Og eventuelt andre lignende spørgsmål som I selv finder på. I bedes venligst angive de websider (og bøger), hvorfra I henter jeres information. God fornøjelse!

# 2 Elementære kodningsteoretiske begreber

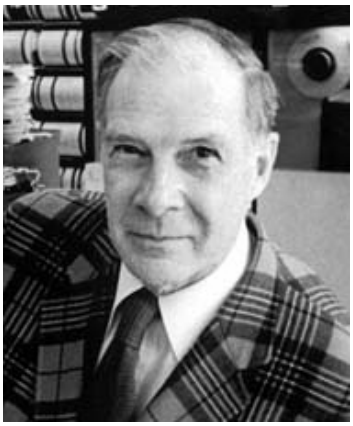
Som antydet i kapitel 1 udgør Shannons 1948-arbejde grundlaget for kodningsteorien. De første fejlkorrigerende koder tilskrives Richard Hamming, eksempelvis den såkaldte $(7, 4)$-Hamming-kode, som Shannon med reference til Hamming selv anvendte i sin artikel.

Hamming var ligesom Shannon ansat ved Bell Telephone Laboratories og det var formentlig derfor, at de to matematikere var bekendte med hinandens arbejde.

## 2.1 Bell Labs

Bell Telephone Laboratories Inc. (Bell Labs) blev grundlagt i 1925 som en afdeling under Western Electric og American Telephone & Telegraph Company (AT&T). Fra 1925 og frem til 1980erne var en af Bell Labs' hovedopgaver at designe det amerikanske telefonnetværk. Western Electric tog sig af at fremstille telefoner, kabler, tavler og andet udstyr, installere dette samt afregne med kunderne. AT&T tog sig af 'long distance'-netværket.

Bell Labs' arbejde bestod af såvel forskning som ingeniørarbejde og udvikling. Forskningsafdelingen, der udgjordes af omkring 12 procent af



Richard Wesley Hamming
(1915-1998)

Hamming studerede matematik ved universiteterne i Chicago og Nebraska frem til 1939. I 1942 modtog han en ph.d. i matematik fra University of Illinois. I 1945 blev Hamming en del af det såkaldte Manhattan-projekt, hvor man arbejdede med udviklingen af atombomben, mere præcist blev han tilknyttet underprojektet kaldet Los Alamos, hvor der blev arbejdet på udviklingen af en brintbombe. I 1946 begyndte Hamming at arbejde for Bell Telephone Laboratories, sammen med blandt andet Shannon. Herfra udgav han i 1950 sin berømte artikel om kodningsteori. I 1956 arbejdede Hamming på den tidlige 650 IBM-computer, og hans arbejde hermed førte blandt andet til udviklingen af højniveau programmeringssprog. I 1976 accepterede Hamming en stilling i datalogi ved Naval Postgraduate School i Monterey. Hamming modtag i sine sene leveår adskillige priser, ikke mindst for sit arbejde inden for kodningsteori. Hamming var iøvrigt kendt for sin røde skotskternede sportsjakke og sine efter sigende vandede vittigheder. (Davis; 2005) `http://www-groups.dcs.st-and.ac.uk/~history/Mathematicians/Hamming.html`

Bell Labs' tekniske personel, var den afdeling der var ophav til de tusindvis af opfindelser og opdagelser som har været med til at forme vore dages kommunikationssystemer. Den vigtigste af disse er formentligt transistoren (1947) uden hvilken der ikke ville være nogen moderne elektronik såsom PC'er, CD-afspillere og så videre. Derudover kan eksempelvis nævnes laseren (1958), solcelle-batteriet (1954), laboratoriets arbejde med satellitter og mobiltelefoni, computer operativsystemet UNIX (1969) som er grundlaget for internettet samt programmeringssproget C (1972). Af mindre kendte opfindelser kan nævnes video-telefonen fra 1960erne som aldrig rigtigt slog an. Igennem tidernes løb har forskere ved Bell Labs modtaget seks Nobel-priser i fysik, et hav af andre priser inden for fysik, ingeniørvidenskab og teknologi og i 2006 modtog Bell Labs sågar en GRAMMY, i kategorien teknik vel at mærke, for sit pionerbidrag til lydoptagelser og digital kommunikation[1]. Rent faktisk havde Bell Labs i mange år også deres egen interne matematikuddannelse til kandidatniveau, altså hvad der svarer til en universitetsuddannelse.

Shannon og Hamming, såvel som Golay som vi skal møde senere, var altså en del af Bell Labs' forskningsafdeling. Hamming arbejdede med Bell Labs' computere, ligesom han havde gjort det med Los Alamos' computere (se billedtekst side 15), samt udvikling af de tidlige programmeringssprog – et arbejde der i datalogiens barndomsår blev udført af matematikere. Det var dette arbejde med computerberegninger der inspirerede Hamming til at udvikle sine fejlkorrigerende koder. Hamming selv siger om dette:

> Forfatteren blev ledt på sporet af det i denne artikel beskrevne studie på baggrund af en betragtning af store beregningsmaskiner i hvilke et stort antal af operationer må udføres uden en eneste fejl i det endelige resultat. (Hamming; 1950, oversat fra engelsk, side 147)

Tidligere havde man udelukkende benyttet sig af koder som kunne opdage fejl, såkaldte *fejldetekterende* koder, men Hamming tog disse koder et skridt videre fra kun at kunne opdage fejl til at kunne rette dem.

Da Hamming begyndte ved Bell Labs i 1946 fandtes der en *Model V* af Stibitz' *Model I* fra 1937. *Model V* var den største i serien, forstået såvel kapacitetsmæssigt som rent fysisk. Computeren fyldte et område på lidt over 300 kvadratmeter (cirka 1.000 kvadratfod) og vejede 10 tons. Beregningsmæssigt svarede computerens kapacitet til hvad der i dag kan findes i en mellemstor lommeregner. Beregninger som det dengang tog timer at løse klares ofte i dag af en almindelig PC'er på få sekunder.

Flere af de begreber som i dag anvendes inden for kodningsteorien er da også navngivet efter Hamming, herunder blandt andet Hamming-afstand og Hamming-kugle. Disse begreber skal vi se nærmere på i dette samt de følgende kapitler.

---

[1] I 1984 blev Bell Labs overtaget af AT&T og siden har det levet en omflakkende tilværelse. I 1995 blev AT&T delt op i tre nye firmaer og som et resultat af dette blev Bell Labs delt i to, således at AT&T Corporation kom til at eje en del kaldet AT&T Research og Lucent Technologies en anden kaldet Bell Laboratories Research. Informationerne i dette afsnit stammer fra `http://www.bellsystemmemorial.com/belllabs.html` og `http://www.bell-labs.com/about/history/`

## 2.2 Om blokkoder

Vi skal begrænse os til at se på den klasse af koder der kaldes *blokkoder* – hvorfor koderne netop kaldes således skal vi vende tilbage til. Ideen er, at vi begynder med at have en mængde af binær information, som vi ønsker at *indkode*. Mængden af binær information består af et antal *dataord*. Et dataord er et talsæt, det vil sige $(a_1, a_2, \ldots, a_k)$, af binære symboler – altså en binær streng af en fastsat længde. En indkodning består i at transformere sådanne talsæt til større talsæt, $(x_1, x_2, \ldots, x_k, x_{k+1}, \ldots, x_n)$, af binære symboler – altså, at vi tilføjer ekstra binær information. De større talsæt kaldes *kodeord*. Samlingen af kodeord udgør blokkoden. Lad os se et eksempel.

**Eksempel 2.1**
Antag, at vi har en mængde af binær information bestående af følgende fire dataord

$$\{00, 01, 10, 11\},$$

som vi ønsker at indkode. Hvert af disse dataord består af to binære symboler ($k = 2$). Indkodningen består nu i at opstille følgende korrespondance:

$$\begin{aligned} 00 &\mapsto 000000 \\ 01 &\mapsto 000111 \\ 10 &\mapsto 111000 \\ 11 &\mapsto 111111, \end{aligned}$$

hvor de binære strenge til højre udgør kodeordene. Vores blokkode, som vi vil kalde $\mathcal{C}$ er da givet ved

$$\mathcal{C} = \{000000, 000111, 111000, 111111\}.$$

$\diamond$

Indkodningsproceduren består altså i at opstille en korrespondance mellem dataordene og kodeordene og benytte denne i transmissionen. (For en god ordens skyld skal det selvfølgelig nævnes at 'modtageren' kender alle de mulige kodeord der anvendes i transmissionen.) For en blokkode må korrespondancen være af en sådan art, at et dataord knyttes til præcis ét kodeord og et kodeord til præcis ét dataord. Med udgangspunkt i ovenstående eksempel kan vi med andre ord også sige, at blokke af to datasymboler hver for sig giver anledning til blokke af seks kodesymboler, heraf følger navnet blokkode. En oplagt måde at indkode et dataord på er ved, at lade de første $x_1, \ldots, x_k$ symboler i kodeordet være udgjort af selve dataordet og de resterende $x_{k+1}, \ldots, x_n$ af den ekstra binære information. Men som eksemplet ovenfor viser behøver det ikke nødvendigvis være sådan. Bemærk, at hvis det ikke på forhånd er klart om et givet talsæt er et data- eller et kodeord, vil vi undertiden blot referere til det som et *ord*.

Fra nu af vil vi altså interessere os for data*ord*, kode*ord* og blokkoder, som vi tilmed nu kan give en generel definition af:

**Definition 2.2**
*En binær* blokkode $\mathcal{C}$ *af størrelse* $m$ *og* bloklængde $n$ *er en mængde af* $m$ *talsæt hvert bestående af* $n$ *symboler fra det binære alfabet. Talsættene kaldes* kodeord.

**Eksempel 2.3**
Vores kode $\mathcal{C}$ fra eksempel 2.1 er altså en binær blokkode bestående af $m = 4$ kodeord hver med bloklængde $n = 6$. ◇

Vi vil nøjes med at se på koder, hvor $m$ er af en bestemt type, nemlig dem hvor $m = 2^k$. En sådan kode kaldes en $(n, k)$-kode. For sådanne koder er det praktisk at tænke på dataord som binære ord af længde $k$ (dem findes der $2^k$ af) og kodeord som binære ord af længde $n$.

**Eksempel 2.4**
Vores kode $\mathcal{C}$ fra eksempel 2.1 har som set $k = 2$ og derfor $m = 2^k = 2^2 = 4$. Da koden har bloklængde $n = 6$ er der således tale om en $(6, 2)$-kode. ◇

Forholdet mellem antallet af binære symboler i en blokkodes data- og kodeord kendes som *informationsraten*.

**Definition 2.5**
Informationsraten *for en* $(n, k)$*-blokkode er defineret som forholdet* $k/n$.

**Eksempel 2.6**
Informationsraten for vores kode $\mathcal{C}$ fra før er $2/6 = 1/3$ svarende til, at hver gang vi transmitterer 6 symboler, er det kun 2 symboler, som er egentlige data. ◇

Informationsraten er dermed et udtryk for, hvor mange symboler vi skal sende for at vi får ét datasymbol igennem. I ovenstående eksempel skulle vi således sende tre symboler for at vi fik ét datasymbol igennem.

Det er klart, at man umiddelbart er interesseret i en høj informationsrate svarende til meget data per transmitteret symbol. Der er imidlertid også andre vigtige egenskaber at tage hensyn til (ellers kunne man jo bare lade være med at kode dataene). Som beskrevet tidligere er formålet med at indkode dataord til kodeord inden de transmitteres, at gøre det muligt i den efterfølgende afkodning at korrigere for eventuelle fejl opstået som følge af støj på kanalen. For, så at sige, at gøre afkodningen mere sikker er man interesseret i at kodeordene ikke 'ligner' hinanden alt for meget, da dette nemt kan give anledning til afkodningsfejl. Til dette formål indføres begrebet *Hamming-afstand*.

**Definition 2.7: Hamming-afstand**
Hamming-afstanden $d(\mathbf{x}, \mathbf{y})$ *mellem to binære talsæt* $\mathbf{x}$ *og* $\mathbf{y}$ *af længde* $n$ *er lig antallet af pladser, hvor* $\mathbf{x}$ *og* $\mathbf{y}$ *er forskellige.*

Med begrebet afstand i denne sammenhæng tænker vi ikke på den almindelige geometriske, også kaldet den euklidiske, afstand mellem to punkter. Derimod er der tale om et mere 'generaliseret afstandsbegreb' som giver

et mål for, hvor 'tæt' to kodeord er på hinanden, eller 'hvor meget de ligner' hinanden. (Brugen af bogstavet $d$ til angivelsen af afstand skyldes det engelske ord 'distance'.)

**Eksempel 2.8**
De følgende to kodeord har Hamming-afstand 5, idet de er forskellige på 5 pladser:

$$0100100110011101$$
$$0110\underline{0}0011\underline{1}01\underline{0}01$$

◇

Vi skal se lidt nærmere på sådanne generaliserede afstandsbegreber, som Hamming-afstanden er et eksempel på, senere i dette kapitel. Et andet begreb som det også er relevant at se på i forbindelse med Hamming-afstanden er *vægten* af et kodeord.

**Definition 2.9: Vægt**
Vægten $w(\mathbf{x})$ *af et kodeord* $\mathbf{x}$ *defineres som antallet af symboler i kodeordet forskellige fra 0.*

For binære kodeord, som er dem vi beskæftiger os med, svarer vægten altså til antallet af 1-taller i kodeordet. (Brugen af $w$ skyldes her det engelske ord 'weight'.)

**Eksempel 2.10**
For kodeordene i eksempel 2.1 er de tilhørende vægte $0, 3, 3, 6$. Hamming-afstanden for par af kodeord i koden er enten 6, da d(000000,111111)=6, eller 3, hvilket gør sig gældende for de resterende afstande. Hvorfor? ◇

Det ses endvidere, at Hamming-afstanden mellem nul-kodeordet, kaldet $\mathbf{0}$, det kodeord der består udelukkende af nuller, og et andet kodeord $\mathbf{x}$ er lig vægten af $\mathbf{x}$.

Hamming-afstanden opfylder fire egenskaber, der generelt altid kræves for at man i matematikken benytter begrebet 'afstand'.

**Sætning 2.11**
*For vilkårlige binære talsæt* $\mathbf{x}$, $\mathbf{y}$ *og* $\mathbf{z}$ *gælder:*

*i.* $d(\mathbf{x}, \mathbf{y}) > 0$ *hvis* $\mathbf{x} \neq \mathbf{y}$.
*ii.* $d(\mathbf{x}, \mathbf{x}) = 0$.
*iii.* $d(\mathbf{x}, \mathbf{y}) = d(\mathbf{y}, \mathbf{x})$.
*iv.* $d(\mathbf{x}, \mathbf{y}) \leq d(\mathbf{x}, \mathbf{z}) + d(\mathbf{y}, \mathbf{z})$.

**Bevis**
De første tre betingelser indses (forholdsvist) nemt, hvorfor beviserne for disse er henlagt til opgave 25. Den fjerde, også kendt som *trekantsuligheden*, må vi have et bevis for.

Bemærk, at $d(\mathbf{x}, \mathbf{y})$ er det mindste antal af ændringer af bit der kræves for at ændre $\mathbf{x}$ til $\mathbf{y}$. Bemærk dernæst, at vi også kan ændre $\mathbf{x}$ til $\mathbf{y}$ ved først at foretage $d(\mathbf{x}, \mathbf{z})$ ændringer, og derved lave $\mathbf{x}$ om til $\mathbf{z}$, og dernæst foretage $d(\mathbf{z}, \mathbf{y})$ ændringer, og derved lave $\mathbf{z}$ om til $\mathbf{y}$. Men det betyder jo netop, at $d(\mathbf{x}, \mathbf{y}) \leq d(\mathbf{x}, \mathbf{z}) + d(\mathbf{y}, \mathbf{z})$. □

## 2.3 Afkodning med 'nærmeste nabo'

En mulig procedure til afkodning af et modtaget ord $\mathbf{v}$ er at afkode til et dataord, hvis kodeord $\tilde{\mathbf{v}}$ har den mindste Hamming-afstand til $\mathbf{v}$. En sådan afkodning kaldes en *'nærmeste nabo'*-afkodning.

**Eksempel 2.12**
Lad os igen tage udgangspunkt i eksempel 2.1, hvor vi har givet koden $\mathcal{C} = \{000000, 000111, 111000, 111111\}$.

Antag, at vi efter en transmission har modtaget ordet $\mathbf{v} = 011000$. Vi kan nu udregne Hamming-afstanden mellem $\mathbf{v}$ og de fire kodeord i $\mathcal{C}$. Vi får, at $d(\mathbf{v}, 000000) = 2$, $d(\mathbf{v}, 000111) = 5$, $d(\mathbf{v}, 111000) = 1$ og $d(\mathbf{v}, 111111) = 4$. Med 'nærmeste nabo'-afkodning skal vi altså afkode det modtagne ord $\mathbf{v}$ til kodeordet 111000. ◇

Et vigtigt begreb i forbindelse med 'nærmeste nabo' afkodning er en kodes såkaldte minimumsafstand. Vi skal definere begrebet her og vende tilbage til brugen af det i næste kapitel.

**Definition 2.13: Minimumsafstand**
*Lad $\mathcal{C} = \{\mathbf{x}_1, \ldots, \mathbf{x}_m\}$ med $m \geq 2$ være en kode. Da er* minimumsafstanden $d_{\min}$ *for $\mathcal{C}$ defineret som den mindste af Hamming-afstandene mellem samtlige par af forskellige kodeord. Med symboler kan dette skrives som*

$$d_{\min} = \min_{\mathbf{x}_i, \mathbf{x}_j \in \mathcal{C}, i \neq j} d(\mathbf{x}_i, \mathbf{x}_j),$$

*hvor $i, j \in \{1, 2, \ldots, m\}$ og* min *betyder at vi tager den mindste (minimum).*

Da $\mathbf{x}_i \neq \mathbf{x}_j$ må vi have, at $d_{\min} \geq 1$, altså at minimumsafstanden af en kode altid må være mindst 1. En $(n, k)$-blokkode med minimumsafstand $d_{\min}$ kendes i litteraturen også som en $(n, k, d_{\min})$-blokkode.

**Eksempel 2.14**
For vores kode $\mathcal{C}$ er minimumsafstanden den mindste af Hamming-afstandene mellem par af kodeord, det vil sige 3 (jævnfør eksempel 2.10). ◇

For en kode med en stor minimumsafstand skal der opstå mange fejl, før man ved 'nærmeste nabo'-afkodning kommer så langt væk fra det rigtige kodeord, at man får en afkodningsfejl. Af denne årsag er man generelt interesseret i at benytte koder med stor minimumsafstand. Jo større bloklængde en kode har, jo bedre er mulighederne for at få en

stor minimumsafstand, fordi der er flere positioner på hvilke, kodeordene kan være forskellige – vel at mærke samtidig med at vi holder $k$ så lav som muligt, således at vi giver os selv den nødvendige 'plads' $(n - k)$ til at korrigere for eventuelle fejl. På den anden side er vi dog heller ikke interesseret i at have alt for meget af denne 'plads', idet vi jo ikke ønsker at sende mere ekstra binære information end højst nødvendigt. Vi har altså to modsatrettede interesser:

1. Vi vil gerne have stor minimumsafstand, det vil sige, at $n$ skal være stor, og at $k$ skal være lille.
2. Vi vil gerne have høj informationsrate, det vil sige, at $n$ ikke må være stor i forhold til $k$.

Et af hovedproblemerne inden for kodningsteori er at finde koder, som så vidt muligt tilfredsstiller ovenstående behov. Samtidigt vil man gerne have at koderne skal være simple at ind- og afkode.

## 2.4 Generaliserede afstandsbegreber

Et generaliseret afstandsbegreb der opfylder de i sætning 2.11 fire givne betingelser kaldes også en *metrik*. (Hamming-afstanden er altså en metrik.) Formuleringen af en metrik som vi kender den i dag skyldes den franske matematiker Maurice Fréchet (1878-1973) som i 1906 publicerede en afhandling, hvori et sådant afstandsbegreb indgik. Selve navnet metrik skyldes dog ikke Fréchet, men derimod den tyske matematiker Felix Hausdorff (1868-1942).

Fréchets motivation for indførelsen af et generaliseret afstandsbegreb var, at han ønskede at studere såkaldte abstrakte rum eller *punktmængder* opfattet som rum. En mængde af punkter i gængs forstand kan eksempelvis være en mængde af punkter fra planen angivet ved deres koordinatsæt. Når man taler om sådanne punktmængder som *rum* er man ikke blot interesseret i de specifikke punkter, men også i hvordan disse punkter relaterer sig til hinanden. I planen er punkter eksempelvis relateret ved deres euklidiske afstand. Fréchet bemærkede imidlertid, at det ikke kun er den euklidiske afstandsfunktion der kan fungere som afstand og han definerede derfor en mere generel afstand – en *metrik*. Bemærk, at den euklidiske afstand i sig selv er et eksempel på en metrik, men at en metrik altså er et mere generelt (generaliseret) afstandsbegreb. Man siger, at en mængde (af punkter) sammen med en metrik udgør et *metrisk rum*. (Mængden bestående af forskellige talsæt af længde $n$, svarende til punkter, sammen med Hamming-afstanden udgør således et metrisk rum.)

Afstandsbegrebet kan generaliseres endnu yderligere ved brug af begrebet *omegn*, hvilket Hausdorff gjorde i 1914. Anvendelsen af omegnsafstandsbegrebet i stedet for en metrik giver således også anledning til et mere generelt rum end det metriske, et såkaldt *topologisk rum* eller et *Hausdorff-rum*. Hausdorff viste, hvilket vi ikke skal, at ethvert metrisk rum er et topologisk rum. Man siger i dag, at en mængde af punkter sammen med en *topologi* (omegn) udgør et topologisk rum. Ordet topologi kommer af det græske ord *topos* hvilket betyder *sted* og altså er at sammenligne med omegn. Hausdorff generaliserede

også i termer af omegne kontinuitetsbegrebet til funktioner i topologiske rum og viste en række relaterede resultater. Med matematikhistorikeren Victor Katz' ord viste Hausdorff, at »et topologisk rum er et naturligt miljø at studere de klassiske resultater om funktioner af en variabel i« (Katz; 1998, side 819, oversat fra engelsk).

Selv om Fréchet oftest regnes for faderen til afstandsbegrebet svarende til en metrik har senere matematikhistoriske studier vist, at oprindelsen måske skal findes en smule tidligere i historien, nemlig hos den (litauisk fødte) tyske matematiker Hermann Minkowski (1864-1909).

## 2.5 Opgaver

**Opgave 20**
Hvad karakteriserer en binær $(n, k)$-blokkode? Hvad siger informationsraten for koden noget om? Konstruer binære blokkoder med følgende informationsrater, hvor $m = 2^k$:
a. $^2/_5$.
b. $^3/_8$.
c. $^4/_5$.
d. $^4/_7$.
Angiv de forskellige Hamming-afstande og vægte for de konstruerede koder i spørgsmål (a) og (b). Angiv minimumsafstanden, $d_{\min}$, og den mindste vægt i samtlige af de fire konstruerede koder.

**Opgave 21**
Bestem $n$, $m$, og $d_{\min}$ for følgende koder:
a. $\mathcal{C} = \{000, 010, 011\}$.
b. $\mathcal{C} = \{00011, 00101, 11101, 11000\}$.
c. $\mathcal{C} = \{0000, 1100\}$.
d. $\mathcal{C} = \{00, 01, 10, 11\}$.

**Opgave 22**
I følgende tilfælde ønskes en kode med de givne parametre enten konstrueret eller der ønskes argumenteret for, at en sådan kode ikke eksisterer:
a. $n = 8$, $m = 2$ og $d_{\min} = 8$.
b. $n = 8$, $m = 3$ og $d_{\min} = 8$.
c. $n = 3$, $m = 9$ og $d_{\min} = 1$.
d. $k = 3$, $n = 4$ og $d_{\min} = 2$.
e. $k = 3$, $n = 4$ og $d_{\min} = 3$.
f. $n = 4$, $m = 8$ og $d_{\min} = 2$.
g. $n = 4$, $m = 8$ og $d_{\min} = 0$.
h. $k = 3$, $n = 4$ og mindste vægt lig 0.
i. $n = 5$, $m = 3$ og $d_{\min} = 4$.
j. $n = 5$, $m = 32$ og mindste vægt lig 2.
k. $n = 13$, $m = 2$ og mindste vægt lig 7.

**Opgave 23**
Vis, at der for tre talsæt $\mathbf{u}$, $\mathbf{v}$ og $\mathbf{z}$ gælder for Hamming-afstanden, at

$$d(\mathbf{u}, \mathbf{v}) = d(\mathbf{u} + \mathbf{z}, \mathbf{v} + \mathbf{z}),$$

ved først at se på tilfældet, hvor $u_i = v_i$ og dernæst på tilfældet hvor $u_i \neq v_i$ for $i = 1, \ldots, n$.

**Opgave 24**
For en binær blokkode givet ved

$$\mathcal{C} = \{11100, 01001, 10010, 00111\}$$

ønskes afkodning ved 'nærmeste-nabo' anvendt til at afkode følgende modtagne kodeord:
a. 10000.
b. 01100.
c. 01001.
d. 00100.

**Opgave 25**
Argumenter for, at betingelserne *i*, *ii*, *iii* i sætning 2.11 gælder.

**Opgave 26 (Essay-opgave)**
Søg på 'History of Bell Labs' og andre lignende søgeord på Internettet og se hvilke informationer I kan finde. Kan I finde nogle informationer som linker Bell Labs til Shannon og Hamming? (Eventuelt ved at søge på 'Shannon and Bell Labs' og 'Hamming and Bell Labs'.)

Forsøg med disse informationer samt dem i undervisningsmaterialet at give bud på svar til følgende spørgsmål:
a. Hvad kan man sige om Bell Labs, når man ved at deres forskningsafdeling udgjordes af omkring 12 procent af institutionens samlede tekniske personel? Er 12 procent meget eller lidt i denne sammenhæng?
b. Kan man sige noget om, hvorvidt Bell Labs udførte grundforskning eller kun var interesserede i praktiske anvendelser?

Og eventuelt andre lignende spørgsmål som I selv finder på. Igen bedes I venligst angive de eventuelle websites I henter jeres informationer fra.

Som antydet i dette kapitel var Bell Labs en kæmpe forskningsinstitution i 1940'erne, 1950'erne, 1960'erne og 1970'erne i USA. Specielt i årene lige efter 2. verdenskrig og også et stykke op i den kolde krig var der en stor strøm af penge fra den amerikanske stat, herunder militære institutioner, til forskningsprojekter såvel på universiteter som i private virksomheder som Bell Labs.
c. Hvad tror I dette har betydet for den naturvidenskabelige, herunder den matematiske, forskning generelt? Og specielt for Bell Labs?

**Opgave 27 (Essay-opgave)**
I sin artikel fra 1950 introducerer Hamming sin afstandsfunktion, den vi i dag kender som Hamming-afstanden, på følgende vis:

> I vores rum bestående af $2^n$ punkter introducerer vi en afstand, eller som det normalt kaldes, en *metrik*, $D(x, y)$. Definitionen af denne metrik er baseret på observationen af at en enkelt fejl i et kode-punkt ændrer en koordinat, to fejl to koordinater og generelt set producerer $d$ fejl en forskel i $d$ koordinater. Altså kan vi definere afstanden $D(x, y)$ mellem to punkter $x$ og $y$ som antallet af koordinater for hvilke $x$ og $y$ er forskellige. [...] Afstandsfunktionen opfylder de tre gængse betingelser for en metrik, nemlig
>
> $$\begin{aligned} D(x,y) &= 0 \text{ hvis og kun hvis } x = y \\ D(x,y) = D(y,x) &> 0 \text{ hvis } x \neq y \\ D(x,y) + D(y,z) &\geq D(x,z) \text{ (trekantsulighed) .} \end{aligned}$$
>
> (Hamming; 1950, side 11, oversat fra engelsk)

Hammings begreb 'kode-punkt' svarer til et kodeord.

a. Redegør for at Hammings definition af en metrik her ovenfor er den samme som definitionen af Hamming-afstanden i definition 2.7.
b. Redegør for at de betingelser Hammings metrik her ovenfor opfylder er identiske med de i sætning 2.11 givne.

Som berettet om i afsnit 2.4 introducerede Hausdorf det generaliserede afstandsbegreb en topologi (omegn) og de dertil hørende topologiske rum. I termer af dette ny generelle afstandsmål var Hausdorff dernæst i stand til at generalisere klassiske begreber og resultater om funktioner i én variabel til topologiske rum. Hausdorff studerede altså disse klassiske resultater i nogle nye omgivelser eller som Katz formulerer det »et naturligt miljø«. Eller man kan sige, at det generelle afstandsbegreb en toplogi udgjorde en ny *behandlingsmåde* til at studere funktioner i én variable med.

c. Med udgangspunkt i ovenstående beskrivelse af en *behandlingsmåde* kan der altså drages paralleller til Hammings brug af det generaliserede afstandsbegreb en metrik. På hvilken måde tjener 'metrik' som en behandlingsmåde i ovenstående Hamming-citat? Hvilken problemstilling hjælper dette generelle afstandsbegreb med at behandle?

# 3 Fejldetektion versus fejlkorrektion

Som beskrevet i begyndelsen af forrige kapitel anvendte man i midten af 1940'erne på Bell Labs udelukkende fejldetekterende koder. Det betød, at man når man foretog beregninger på computerne var i stand til at identificere hvilke resultater der var påhæftet med fejl. Desværre gik der meget computertid tabt, idet en computer ville stoppe sine beregninger når en fejl blev detekteret. I sin 1950-artikel siger Hamming selv følgende om dette:

> Den selv-checkende egenskab betød at disse fejl ikke introducerede uopdagede fejl. Da maskinerne blev kørt uden overvågning om natten og i weekenderne betød fejlene imidlertid at beregningerne ofte blev bragt til standsning [...] Forekomsten af isolerede fejl, selv når disse blev opdaget, kan imidlertid intervenere alvorligt med den normale brug af sådanne maskiner. Altså forekommer det ønskværdigt at undersøge det næste skridt ud over fejldetektion, nemlig fejlkorrektion. (Hamming; 1950, side 147, oversat fra engelsk)

Med udgangspunkt i hvad Hamming således selv skriver i 1950, lader hans motivation for at introducere fejlkorrigerende koder altså til at stamme fra et ønske om at effektivisere brugen af Bell Labs' computere og de beregninger som blev udført på disse – og muligvis kan man også spore en irritation over den spildtid der var involveret i proceduren hidtil. I et interview som matematikhistorikeren Thomas Thompson foretog med Hamming i 1977 fortæller Hamming mere om denne irritation. Det forholdt sig således, at *Model V* computeren havde to indstillinger i forbindelse med en eventuel fejldetektion; 'dag' og 'nat/weekend'. Når maskinen stod på 'dag', ville en fejldetektion give sig til udslag i, at maskinen stoppede og en alarm gik igang. De tilstedeværende operatører kunne derefter ved hjælp af et lyspanel gå igang med at lokalisere fejlen, korrigere den og sætte beregningen igang igen. Stod maskinen derimod på 'nat/weekend' ville en fejldetektion medføre at maskinen stoppede, droppede den pågældende beregning og påbegyndte en anden. Den oprindelige beregning måtte så køres igen på et senere tidspunkt. Hamming havde ikke førsteprioritet over computeren, hvorfor hans beregninger typisk blev kørt i weekenden. Om denne uheldige omstændighed fortalte Hamming til Thompson i 1977:

> To weekender i træk kom jeg ind og opdagede, at alle mine beregninger var blevet droppet og at der intet var blevet lavet. Jeg var virkelig ophidset og irriteret, fordi jeg ville have de svar

> og to weekender var gået tabt. Og så sagde jeg: »For fanden, hvis maskinen kan opdage en fejl, hvorfor kan den så ikke lokalisere positionen af fejlen og rette den?« (Thompson; 1983, side 17, oversat fra engelsk)

Efterhånden som Hamming begyndte at arbejde seriøst med sine fejlkorrigerende koder må vi dog formode, at han, udover at være drevet af sin irritation, også er blevet yderligere motiveret af de matematiske strukturer som koderne indeholder og bygger på.

## 3.1 Et eksempel på en fejldetekterende kode

Et eksempel på en af disse fejldetekterende koder som Hamming omtaler ovenfor er den såkaldte *paritetscheckkode*. Indkodning af et dataord, som består af $k$ bit, foregår ved at dataordet tilføjes én ekstra bit, således at antallet af 1-taller i kodeordet er lige[1]. Antallet af binære symboler i kodeordet, $n$, er altså lig $k+1$, det vil sige $k = n-1$, hvorfor en paritetscheckkode er en $(n, n-1)$-blokkode.

**Eksempel 3.1**
Et konkret eksempel på en paritetscheckkode kan være den hvis dataord består af fire bit. Disse indkodes til kodeord bestående af fem bit på følgende vis:

$$\begin{array}{ccc} 0000 & \mapsto & 00000 \\ 0001 & \mapsto & 00011 \\ 0010 & \mapsto & 00101 \\ \vdots & & \vdots \\ 1111 & \mapsto & 11110 \end{array}$$

Koden kan detektere én fejl, men ikke korrigere nogen. Til gengæld har koden en høj informationsrate, nemlig $(n-1)/n$, i dette tilfælde $4/5$. $\diamond$

Der gælder helt generelt, at paritetscheckkoder har minimumsafstand 2. Hvorfor?

Paritetscheckkoder regnes også for at være såkaldte *trivielle* koder, idet indkodningsproceduren for disse er så banal, som den nu engang er. Koder bestående af blot ét kodeord eller alle talsæt af en given længde regnes også for trivielle koder. Sådanne koder er i praksis ikke særligt anvendelige: Hvis man kun har ét kodeord kan man kun sende én bestemt information og i det tilfælde kan man jo lige så godt lade være med at kode beskeden, for lige meget hvad der modtages, så vil det være den besked der er sendt. Hvis en kode udgøres af alle talsæt af en given længde $n$, det vil sige $m = 2^n$, så vil man i tilfælde af støj på kanalen med garanti

[1] Fastsættes antallet af 1-taller i koden til at være et ulige tal kaldes koden for en *ulige* paritetscheckkode. Den her beskrevne kode er en *lige* paritetscheckkode. Ordet paritet kan da også oversættes til 'lige-ulighed'.

få en afkodningsfejl. I modsætning til de trivielle koder er selvfølgelig de *ikke-trivielle* koder, som vi skal se eksempler på i kapitel 4.

## 3.2 $t$-fejldetekterende og $t$-fejlkorrigerende koder

Vi skal nu som overskriften antyder gå lidt dybere ned i beskrivelsen af, hvordan koder kan bruges til at opdage og rette fejl. Men lad os begynde med et eksempel.

**Eksempel 3.2**
Igen kigger vi på vores kode $\mathcal{C} = \{000000, 000111, 111000, 111111\}$ fra tidligere.

Antag, at vi vil transmittere kodeordet 000111. Da vil de ord som afkodes korrekt med 'nærmeste nabo' til 000111 hos modtageren være de følgende: 100111; 010111; 001111; 000011; 000101; 000110. Altså de modtagne ord hvor der maksimalt er fejl på én plads. Men hvad sker der hvis der er to eller flere fejl i det modtagne ord? Antag for eksempel at der er fejl på de første to pladser, således at det modtagne ord er 110111. I termer af Hamming-afstand ligger dette ord tættere på kodeordet 111111 end på kodeordet 000111, hvorfor vi med 'nærmeste nabo'-afkodning vil få en afkodningsfejl. ◇

Med udgangspunkt i dette eksempel kan vi give en definition for, hvornår en kode siges at kunne rette eller *korrigere* $t$ fejl.

**Definition 3.3: $t$-fejlkorrigerende**
*Lad $t$ være et positivt heltal. En kode $\mathcal{C}$ siges at være* $t$-fejlkorrigerende*, hvis 'nærmeste-nabo'-afkodning er i stand til at korrigere $t$ eller færre fejl.*

I definition 3.3 antager vi, at der i tilfælde af flere kandidater til 'nærmeste-nabo'-afkodningen vil blive rapporteret et afkodningssvigt. Koden $\mathcal{C}$ i ovenstående eksempel er altså i stand til at rette én fejl og er dermed en 1-fejlkorrigerende kode.

Selv om vi ikke altid er i stand til at rette fejl i et modtaget kodeord kan det stadig være relevant at vide, at der forekommer fejl i kodeordet. Vi giver følgende definition for, hvornår en given kode siges at kunne opdage eller *detektere* $t$ fejl:

**Definition 3.4: $t$-fejldetekterende**
*Lad $t$ være et positivt heltal. Hvis et kodeord fra en kode $\mathcal{C}$ pådrager sig mindst 1 men højst $t$ fejl og det resulterende talsæt* ikke *bliver til et andet kodeord i $\mathcal{C}$, siges $\mathcal{C}$ at være* $t$-fejldetekterende*.*

**Eksempel 3.5**
$\mathcal{C} = \{000000, 000111, 111000, 111111\}$ er 2-fejldetekterende, da vi ved at ændre et hvilket som helst kodeord på en eller to pladser ikke får et andet kodeord fra $\mathcal{C}$. ◇

Der er altså en grænse for, hvor mange fejl en given kode kan enten korrigere eller detektere og ovenstående eksempler indikerer, at denne grænse hænger sammen med Hamming-afstandene mellem kodeordene. Og en kodes fejldetekterende og -korrigerende egenskaber kan da også udtrykkes mere elegant i termer af den mindste Hamming-afstand for en kode, minimumsafstanden. Faktisk er der tale om, at vi kan vise to sætninger som sammenkæder henholdsvis de fejldetekterende og de fejlkorrigerende egenskaber med minimumsafstanden. Begge disse sætninger er 'hvis og kun hvis' sætninger, hvorfor vi skal vise dem 'begge veje'. Vi begynder med sætningen om fejldetektion.

**Sætning 3.6**
*En kode $\mathcal{C}$ er $t$-fejldetekterende hvis og kun hvis $d_{\min} \geq t+1$.*

**Bevis**
I bund og grund er sætning 3.6 blot en omskrivning af definition 3.4, men vi kan også indse, at sætningen gælder ved hjælp af definitionen.

Antag, at $\mathcal{C}$ er $t$-fejldetekterende. Dette betyder ifølge definitionen, at et kodeord kan pådrage sig mindst 1 og højst $t$ fejl, sålænge at det fejlpåhæftede kodeord ved de $t$ fejl ikke bliver til et andet kodeord. Men dette betyder jo netop, at $d_{\min} - 1 \geq t$.

Antag nu, at $d_{\min} \geq t+1$. Det er det samme som $d_{\min} - 1 \geq t$, hvilket jo betyder at $\mathcal{C}$ kan detektere netop $t$ fejl. □

For at kunne korrigere $t$ fejl ved 'nærmeste nabo'-afkodning kræves imidlertid en større minimumsafstand end for at kunne detektere $t$ fejl.

**Sætning 3.7**
*En kode $\mathcal{C}$ er $t$-fejlkorrigerende hvis og kun hvis $d_{\min} \geq 2t+1$.*

**Bevis**
Antag, at $d_{\min} \geq 2t+1$. Hvis et kodeord $\mathbf{x}$ pådrager sig mellem 1 og $t$ fejl, vil der om det modtagne ord $\mathbf{v}$ gælde, at $1 \leq d(\mathbf{x}, \mathbf{v}) \leq t$. Det betyder, at $d(\mathbf{x}, \mathbf{v}) < t+1$, hvorfor vi ifølge antagelsen har, at

$$d(\mathbf{x}, \mathbf{v}) < t+1 \leq d_{\min} - t.$$

For et andet kodeord $\tilde{\mathbf{x}}$ gælder, at $d(\mathbf{x}, \tilde{\mathbf{x}}) \geq d_{\min}$ og da $d(\mathbf{x}, \mathbf{v}) \leq t$ har vi endvidere

$$d(\mathbf{x}, \mathbf{v}) < t+1 \leq d_{\min} - t \leq d(\mathbf{x}, \tilde{\mathbf{x}}) - d(\mathbf{x}, \mathbf{v}).$$

Fra trekantsuligheden, her $d(\mathbf{x}, \tilde{\mathbf{x}}) \leq d(\mathbf{x}, \mathbf{v}) + d(\tilde{\mathbf{x}}, \mathbf{v})$, følger, at

$$d(\mathbf{x}, \mathbf{v}) < t+1 \leq d_{\min} - t \leq d(\mathbf{x}, \tilde{\mathbf{x}}) - d(\mathbf{x}, \mathbf{v}) \leq d(\tilde{\mathbf{x}}, \mathbf{v}),$$

så $\mathbf{x}$ er altså det kodeord, som ligger tættest på $\mathbf{v}$. Det modtagne ord $\mathbf{v}$ vil derfor blive afkodet som det dataord, der hører til $\mathbf{x}$. Koden kan altså rette $t$ fejl.

Den anden vej skal vi vise ved en modstrid. Vi antager, at $\mathcal{C}$ er $t$-fejlkorrigerende, men at der findes to kodeord $\mathbf{x}$ og $\tilde{\mathbf{x}}$ med $d(\mathbf{x}, \tilde{\mathbf{x}}) = d_{\min} \leq 2t$ (bemærk, at dette er det modsatte af $d_{\min} \geq 2t + 1 \Leftrightarrow d_{\min} > 2t$). For at gøre det nemmere for os selv antager vi, at pladserne hvorpå de to kodeords bit afviger er placeret i begyndelsen af disse, altså at

$$x_i \neq \tilde{x}_i \text{ for } i = 1, \ldots, d_{\min} \text{ og } x_i = \tilde{x}_i \text{ ellers.}$$

Lad $\mathbf{v}$ være ordet fastlagt ved

$$v_i = \tilde{x}_i \text{ for } i = 1, \ldots, t \text{ og } v_i = x_i \text{ ellers,}$$

hvor $t < d_{\min}$. $\mathbf{v}$ er altså ordet

$$\mathbf{v} = \underbrace{\tilde{x}_1 \ldots \tilde{x}_t}_{t} \underbrace{x_{t+1} \ldots x_{d_{\min}}}_{d_{\min}-t} \underbrace{\tilde{x}_{d_{\min}+1} \ldots \tilde{x}_n}_{n-d_{\min}}.$$

Så er $d(\mathbf{v}, \mathbf{x}) = t$. Da koden er $t$-fejlkorrigerende er $d(\mathbf{v}, \tilde{\mathbf{x}}) \geq t + 1$ for ellers er $\mathbf{x}$ jo ikke det nærmeste kodeord. Der gælder imidlertid, at $v_i \neq \tilde{x}_i$ på

$$n - t - (n - d_{\min}) = d_{\min} - t \leq 2t - t = t$$

pladser. Men dette er en modstrid da $d(\mathbf{v}, \tilde{\mathbf{x}}) \geq t + 1$. □

Sætning 3.7 er vigtig, da den knytter information om anvendelsen – i form af et antal fejl, som skal kunne korrigeres – sammen med information om kodens matematiske struktur – minimumsafstanden.

**Eksempel 3.8**
Vores efterhånden velkendte kode $\mathcal{C} = \{000000, 000111, 111000, 111111\}$ har som bekendt minimumsafstand $d_{\min} = 3$ (jævnfør eksempel 2.14).

Af sætning 3.6,

$$d_{\min} \geq t + 1 \Leftrightarrow 3 \geq t + 1,$$

følger at $\mathcal{C}$ kan detektere to fejl, altså er en 2-fejldetekterende kode.

Af sætning 3.7,

$$d_{\min} \geq 2t + 1 \Leftrightarrow 3 \geq 2t + 1,$$

følger at $\mathcal{C}$ kan korrigere én fejl, altså er en 1-fejlkorrigerende kode. ◇

I forbindelse med fejl*detektion* og fejl*korrektion* er det vigtigt at påpege, at de nævnte grænser for kodernes ydelse på disse områder kun gælder, når koderne *udelukkende* anvendes til detektion eller korrektion. For en kode med minimumsafstand $2t + 1$, som anvendes til at korrigere $t$ fejl gælder nemlig, at et modtaget ord $\mathbf{v}$, som indeholder flere end $t$ fejl, kan risikere at komme for tæt på et *forkert* kodeord $\tilde{\mathbf{x}}$, altså $d(\tilde{\mathbf{x}}, \mathbf{v}) < t$, og dermed fejlagtigt blive afkodet til $\tilde{\mathbf{x}}$. Dette vil blive opfattet som en succesfuld afkodning. Koden mister altså nogle af sine fejldetekterende egenskaber, hvis den samtidigt anvendes til fejlkorrektion. I praksis anvender man typisk en såkaldt 'blandet strategi' af såvel fejlkorrektion som fejldetektion. Den blandede strategi består overordnet i at skære ned på antallet af fejl som koderne kan korrigere og detektere, sådan at begge dele kan fungere samtidigt.

## 3.3 En geometrisk tilgang til blokkoder

Definitionen af Hamming-afstanden gør det muligt at benytte en mere intuitiv, geometrisk tilgang til blokkoderne.

**Definition 3.9**
*Lad $B$ være mængden af de $2^n$ forskellige binære talsæt som kan dannes med længde $n$. En* Hamming-kugle *i $B$ med radius $r > 0$ og med centrum i et talsæt $\mathbf{b} \in B$ defineres da som mængden af de talsæt i $B$, hvis Hamming-afstand til talsættet $\mathbf{b}$ er højst $r$. Vi kalder denne mængde $S(\mathbf{b}, r)$ og skriver med symboler*

$$S(\mathbf{b}, r) = \{\mathbf{x} \in B \mid d(\mathbf{b}, \mathbf{x}) \leq r\}.$$

(Brugen af $S$ skyldes det engelske ord 'sphere'.) Hamming-kuglerne giver os mulighed for at illustrere betydningen af minimumsafstanden.

**Sætning 3.10**
*For en kode $\mathcal{C}$ med minimumsafstand $2t+1$ gælder, at hvis man for hvert kodeord $\mathbf{x} \in \mathcal{C}$ placerer en Hamming-kugle med radius $t$ centreret i $\mathbf{x}$, altså $S(\mathbf{x}, t)$, så vil disse Hamming-kugler ikke overlappe hinanden.*

**Bevis**
Sætningen vises ved en modstrid. Som set tidligere vil dette sige, at vi antager det modsatte af det vi ønsker at vise og fører denne antagelse til en modstrid.

Antag, at et modtaget ord $\mathbf{v}$ ligger i to forskellige Hamming-kugler, eksempelvis $\mathbf{v} \in S(\mathbf{x}_i, t)$ og $\mathbf{v} \in S(\mathbf{x}_j, t)$. Ifølge trekantsuligheden har vi, at

$$d(\mathbf{x}_i, \mathbf{x}_j) \leq d(\mathbf{x}_i, \mathbf{v}) + d(\mathbf{x}_j, \mathbf{v}),$$

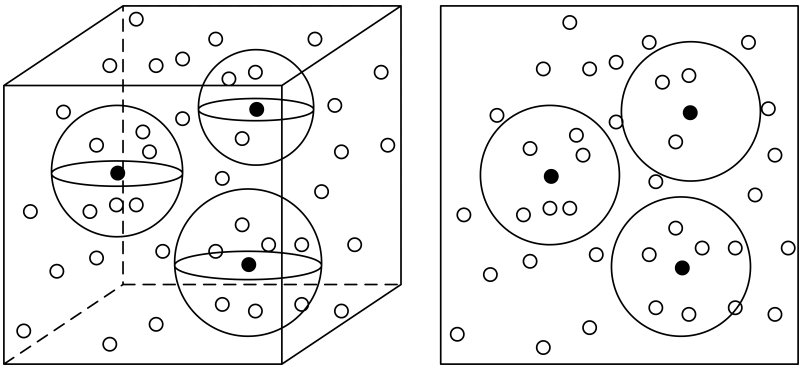
hvilket fører til en modstrid. Dette indses ved, at afstanden fra $\mathbf{v}$ til henholdsvis $\mathbf{x}_i$ og $\mathbf{x}_j$ er højst $t$, mens afstanden mellem de to kodeord $\mathbf{x}_i$ og $\mathbf{x}_j$ i $\mathcal{C}$ er mindst $2t+1$. Altså får vi, at $2t+1$ skal være mindre end eller lig $2t$, hvorfor vi kan konkludere, at $\mathbf{v}$ ikke kan ligge i to forskellige Hamming-kugler og dermed at Hamming-kuglerne er ikke-overlappende. □

Situationen er illustreret på figur 3.1.

Afkodningen for en $t$-fejlkorrigerende kode kan ligeledes forklares ved hjælp af kuglerne. Et modtaget ord, der befinder sig i en Hamming-kugle, afkodes som det dataord, der svarer til kodeordet i Hamming-kuglens centrum.

Figur 3.1 antyder, at man ikke kan være sikker på, at et modtaget ord ligger i en kugle. Alt afhængigt af anvendelsen kan dette afkodningsmæssigt behandles på to måder:

1. En *begrænset afstandsafkoder* afkoder kun modtagne ord, som befinder sig i en Hamming-kugle. Et modtaget ord uden for en sådan afkodes *ikke*. I stedet rapporteres et afkodningssvigt. En sådan afkoder kaldes en *ufuldstændig afkoder*, fordi den ikke afkoder alle modtagne ord.



**Figur 3.1** Hamming-kugler afbildet som henholdsvis rumlige og plane kugler. ∘ angiver vilkårlige talsæt af længde $n$ (mulige modtagne ord) i det pågældende rum, • angiver kodeord.

2. En *fuldstændig afkoder* afkoder derimod ethvert modtaget ord til det nærmeste kodeord. Geometrisk set er der tale om, at afkoderen opdeler 'rummet' mellem kuglerne ved at tildele hvert talsæt bestående af $n$ tal til den kugle, der ligger nærmest. Talsæt med lige stor afstand til flere kugler tildeles vilkårligt. For en $t$-fejlkorrigerende kode gælder, at hvis der modtages et ord, hvor der er opstået *lidt* flere end $t$ fejl, vil dette ord ofte blive afkodet korrekt (der er dog ingen garanti), fordi afstanden mellem kodeordene typisk er større end minimumsafstanden.

Store koder (koder med mange kodeord) har en tendens til at give en alt for kompleks fuldstændig afkoder, og man vælger derfor ofte i praksis at benytte en begrænset afstandsafkoder. Her igennem formindsker man sandsynligheden for at få en *afkodningsfejl* på bekostning af større sandsynlighed for et *afkodningssvigt*.

## 3.4 Opgaver

**Opgave 28**
Opskriv samtlige korrespondancer imellem de 16 talsæt af længde fire og de 16 kodeord i den lige $(5, 4)$-paritetscheckkode.

**Opgave 29**
Opskriv henholdsvis den *lige* og den *ulige* $(4, 3)$-paritetscheckkode.

**Opgave 30**
Konstruer en 3-fejldetekterende kode.

**Opgave 31**
Konstruer en 3-fejlkorrigerende kode.

**Opgave 32**
Bestem ved hjælp af definition 3.4, hvor mange fejl koden givet ved

$$\mathcal{C} = \{11100, 01001, 10010, 00111\}$$

kan detektere.

**Opgave 33**
Bestem ved hjælp af definition 3.3, hvor mange fejl koden givet ved

$$\mathcal{C} = \{111000, 010010, 100100, 001110\}$$

kan korrigere.

**Opgave 34**
For følgende koder ønskes det bestemt, hvor mange fejl de er i stand til at detektere henholdsvis korrigere. (Vink: Anvend sætning 3.6 og sætning 3.7.)
a. $\mathcal{C} = \{000, 010, 011\}$.
b. $\mathcal{C} = \{00011, 00101, 11101, 11000\}$.
c. $\mathcal{C} = \{0000, 1100\}$.
d. $\mathcal{C} = \{00, 01, 10, 11\}$.
e. $\mathcal{C} = \{000000, 111111\}$.
f. $\mathcal{C} = \{00000, 00011, 01111\}$.
g. $\mathcal{C} = \{000, 001, 010, 100, 110, 101, 011, 111\}$.
h. $\mathcal{C} = \{0000, 1010, 0101, 1111\}$.
i. $\mathcal{C} = \{0000, 0011, 1100, 1111\}$.
j. $\mathcal{C} = \{00000000, 00110011, 11001100, 11111111\}$.

**Opgave 35**
Forklar hvorfor en kode vil miste nogle af sine fejldetekterende egenskaber, hvis den samtidig anvendes til fejlkorrektion.

**Opgave 36**
Forklar med dine egne ord hvad der forstås ved begrebet *Hamming-kugle*.

**Opgave 37**
Lad der være givet koden $\mathcal{C} = \{0000, 1111\}$. Bestem hvilke af 16 mulige talsæt af længde fire som vil blive afkodet korrekt for denne kode. Tegn de to Hamming-kugler for denne kode i planen og placer de 16 mulige talsæt af længde fire i forhold til kuglerne.

**Opgave 38**
Lad der være givet koden $\mathcal{C} = \{00000, 11111\}$. Bestem hvilke af 32 mulige talsæt af længde fem som vil blive afkodet korrekt for denne kode. Tegn de to Hamming-kugler for denne kode i planen og placer de 32 mulige talsæt af længde fem i forhold til kuglerne.

# 4 Lineære og perfekte koder

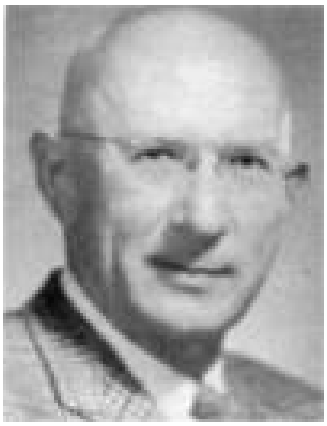
Som nævnt et par gange efterhånden er den første fejlkorrigerende kode vi stifter bekendtskab med i historien den i Shannons 1948-artikel præsenterede $(7, 4)$-Hamming-kode. Denne kode er blot én i en hel familie af Hamming-koder. Omtrent samtidig med at Shannon fik sin artikel publiceret indsendte Hamming også en artikel til *Bell System Technical Journal* (B.S.T.J.), hvori han beskrev sin familie af koder. Hamming fortæller:

> Jeg skrev artiklen [og sendte den til] B.S.T.J. for at få den ud på print og det var det der bremsede den, bang, sådan. Patentafdelingen ville ikke frigive den indtil de havde patentdækning. (Thompson; 1983, side 26-27, oversat fra engelsk)

Hamming selv var skeptisk overfor ideen med at tage patent på et 'stykke matematik' som disse koder:

> Jeg troede ikke på, at de kunne tage patent på en samling matematiske formler. Jeg sagde, de ikke kunne. De sagde, »lagttag os.« De havde ret. Og siden da har jeg vidst, at jeg har meget lidt forståelse af patentlovgivning, fordi, regelmæssigt, ting som du ikke skulle kunne patentere – det er skandaløst – dem kan du tage patent på. (Thompson; 1983, side 27, oversat fra engelsk)

Grundet denne sag om patentering var Hamming ikke i stand til at offentliggøre hele familien af fejlkorrigerende koder, udover $(7, 4)$-koden,



Marcel J. E. Golay (1902-1989)

Golay var oprindelig fra Schweiz og han studerede til elektroingeniør ved ETH i Zürich. Han blev færdig i 1924 og blev umiddelbart derefter ansat ved Bell Laboratories. Fire år senere påbegyndte han en ph.d. i fysik ved University of Chicago. Golay modtog sin ph.d.-grad i 1931 og fik derefter ansættelse ved U.S. Army Signal Corps Laboratories i Fort Monmouth, New Jersey. I 1961 besad Golay en post ved det tekniske universitet i Eindhoven, Holland. Fra 1963 og frem til sin død var han ansat ved Perkin-Elmer company. Igennem hele sit liv var Golay en aktiv videnskabsmand og opfinder, ved sin død ejede han mere end halvtreds patenter. Et af Golays mest betydningsfulde of berømte arbejder er hans en-sides artikel fra 1949 om perfekte koder, herunder de to perfekte Golay-koder. Inden for fysikken opfandt Golay den nu såkaldte 'Golay-celle' til infrarød detektion, hvilken skulle være lige så berømt blandt fysikere som Golay-koderne er det blandt matematikere. (Massey; 1990, side 1-2) `http://www.transpolair.com/routes_polaires/sous_marin/brise_glace.htm`

førend i 1950. På dette tidspunkt havde en anden matematiker, Marcel Golay, på baggrund af Shannons artikel generaliseret $(7, 4)$-Hamming-koden til også at omfatte de andre koden i familien – og publiceret dette. Udover generaliseringen af Hamming-koderne havde Golay også opdaget fire nye koder, i dag kendt som Golay-koder, hvoraf to af disse ligesom Hamming-koderne havde den helt specielle egenskab at de var *perfekte* (dette vil blive forklaret senere i kapitlet). Og ikke nok med det, Golay forudsagde også at der formentlig ikke fandtes andre (ikke-trivielle) perfekte koder end disse – et udsagn der først blev delvist bevist i 1973 af to andre kodningsteoretikere, van Lint og Tietäväinen. Golay præsenterede det hele i en én-sides artikel ved navn *Notes on Digital Coding* – en artikel der siden hen er blevet den nok mest berømte artikel inden for kodningsteori.

I de foregående kapitler har vi betragtet koder som samlinger af talsæt (kodeord) af længde $n$ uden at disse har spillet sammen på nogen speciel måde. For bedre at være i stand til at udtale sig om egenskaber ved koder og for at blive i stand til, i hvert fald i et vist omfang, at kunne konstruere 'rimelige' koder (jævnfør diskussion side 21), skal vi nu indføre det matematikere kalder en *struktur* for de objekter vi ser på.

## 4.1 Definition af nye regneoperationer

For at kunne indføre en sådan struktur må vi indføre nogle nye regneoperationer. I og med at vi udelukkende benytter os af symbolerne 0 og 1 skal regneoperationerne altså gælde for disse. Vi *definerer* da en operation kaldet addition og en operation kaldet multiplikation. Disse er vist i tabel 4.1. Da der ingen forskel er på almindelig multiplikation og vores definerede regneoperation multiplikation markerer vi enten denne med det velkendte symbol $\cdot$ eller ved at sidestille de to størrelser som skal multipliceres, altså $ab = a \cdot b$. Den definerede regneoperation addition afviger imidlertid en smule fra almindelig addition af binære tal, hvorfor vi markerer denne operation med symbolet $\oplus$. Mere præcist er forskellen mellem den almindelige addition og den definerede, at vi i sidstnævnte har $1 \oplus 1 = 0$.

| $\oplus$ | 0 | 1 |
|---|---|---|
| 0 | 0 | 1 |
| 1 | 1 | 0 |

| $\cdot$ | 0 | 1 |
|---|---|---|
| 0 | 0 | 0 |
| 1 | 0 | 1 |

**Tabel 4.1** Definerede regneoperationer for addition og multiplikation.

Man kan uden videre indse, at der med de to nye definerede regneoperationer gælder, at

$$\begin{aligned} a \oplus b &= b \oplus a \\ a \oplus a &= 0 \end{aligned}$$

$$\begin{aligned} 0 \oplus a &= a \\ a \oplus (b \oplus c) &= (a \oplus b) \oplus c \end{aligned}$$

samt

$$\begin{aligned} a(bc) &= (ab)c \\ a(bc) &= ab \oplus ac \\ ab &= ba, \end{aligned}$$

hvor $a$, $b$ og $c$ er enten 0 eller 1. Altså er vi nu i stand til at regne med symbolerne 0 og 1 på en måde der ligner regning med almindelige binære tal, blot er den afgørende forskel at vi her altid har

$$a \oplus a = 0.$$

**Definition 4.1**
*Betragt to kodeord* $\mathbf{x}$ *og* $\mathbf{y}$ *af længde* $n$ *givet ved* $\mathbf{x} = (x_1, x_2, \ldots, x_n)$ *og* $\mathbf{y} = (y_1, y_2, \ldots, y_n)$*. Vi definerer da, at*

$$\begin{aligned} \mathbf{x} + \mathbf{y} &= (x_1 \oplus y_1, x_2 \oplus y_2, \ldots, x_n \oplus y_n) \\ a\mathbf{x} &= (ax_1, ax_2, \ldots, ax_n), \end{aligned}$$

*hvor* $a$*,* $x_i$ *og* $y_i$*, for* $i = 1, 2, \ldots, n$*, er enten 0 eller 1.*

Igen bør vi notere os at vores ny definition af addition her får betydning. Additionen af kodeord som defineret i definition 4.1 er ikke den samme som addition i 2-talssystemet, hvor vi eksempelvis har, at $10 + 11 = 101_2$. Har vi derimod de to kodeord givet ved $(1, 0)$ og $(1, 1)$ vil en addition af disse give os $(0, 1)$. Det er vigtigt at lægge mærke til at $\oplus$ *ikke* optræder imellem de to kodeord $(1, 0)$ og $(1, 1)$, men derimod mellem $1 \oplus 1$ og $0 \oplus 1$. Lad os se yderligere et par eksempler.

**Eksempel 4.2**

$$\begin{aligned} (000111) + (111000) &= (111111) \\ (00110) + (11101) &= (11011) \\ (1100) + (1100) &= (0000) \end{aligned}$$

$\diamond$

Bemærk, at vi i definition 4.1 ikke definerer multiplikation af to kodeord, idet dette ikke er nødvendigt for vores videre fremfærd.

## 4.2 Definition af lineære koder

Med udgangspunkt i ovenstående kan vi nu definere, hvad vi vil forstå ved en såkaldt *lineær* kode. For en lineær kode $\mathcal{C}$ kræves for det første, at

summen af to kodeord i $\mathcal{C}$ også er et kodeord i $\mathcal{C}$ og for det andet, at et kodeord ganget med en skalar (et tal, i vores tilfælde 0 eller 1) ligeledes giver et kodeord i $\mathcal{C}$. Dette kan defineres matematisk på følgende vis:

**Definition 4.3**
*En kode $\mathcal{C}$ kaldes* lineær*, hvis det for to vilkårlige kodeord $\mathbf{x}, \mathbf{y} \in \mathcal{C}$ gælder, at*

*i.* $\mathbf{x} + \mathbf{y} \in \mathcal{C}$ *og*
*ii.* $a\mathbf{x} \in \mathcal{C}$*, hvor $a$ er enten 0 eller 1.*

Rent faktisk er betingelse *ii* en konsekvens af betingelse *i*: Det er jo sådan, at der for at betingelse *ii* er opfyldt for et $\mathbf{x} \in \mathcal{C}$ blot skal gælde, at $\mathcal{C}$ indeholder $\mathbf{x}$, hvilket er forudsat, samt $\mathbf{0}$. Men hvis $\mathcal{C}$ er lineær vil den altid indeholde $\mathbf{0}$, da jo $\mathbf{x} + \mathbf{x} = \mathbf{0}$.

**Eksempel 4.4**
Koden $\mathcal{C} = \{000000, 000111, 111000, 111111\}$ fra eksempel 2.1 er lineær, idet den opfylder betingelserne *i* og *ii* i definition 4.3. ◇

Som indikeret adskillige gange tidligere er minimumsafstanden, $d_{\min}$, en særdeles vigtig parameter for en kode. Selve bestemmelsen af minimumsafstanden kan imidlertid være besværlig, især hvis koden indeholder mange kodeord ($m$ er stor), idet vi altid vil skulle sammenligne det første kodeord med de $m-1$ andre, det andet med de næste $m-2$ andre (da vi jo allerede har sammenlignet det med det første), det tredje med de $m-3$ andre og så videre. Generelt gælder, at vi vil skulle foretage

$$(m-1) + (m-2) + \ldots + 1 = \frac{m(m-1)}{2}$$

sammenligninger af par af kodeord (se også afsnit 4.3). Når vi har at gøre med lineære koder findes der imidlertid en genvej til bestemmelsen af minimumsafstanden. Før vi kan vise den sætning der etablerer denne genvej må vi have en definition og en hjælpesætning, et såkaldt lemma, på banen. Hvis vi ihukommer definitionen af vægten $w(\mathbf{x})$ af et kodeord $\mathbf{x}$ (se definition 2.9) definerer vi, hvad vi vil forstå ved vægten af en kode.

**Definition 4.5**
Vægten af en kode $\mathcal{C}$*, angivet ved $w_{\min}$, er den mindste vægt af alle kodeord i $\mathcal{C}$ forskellige fra $\mathbf{0}$.*

**Lemma 4.6**
*Hvis $\mathcal{C}$ er en binær lineær kode gælder der for alle kodeord $\mathbf{x}$ og $\mathbf{y}$ i $\mathcal{C}$, at $d(\mathbf{x}, \mathbf{y}) = w(\mathbf{x} + \mathbf{y})$.*

**Bevis**
Grundet vores nye definerede regneoperation $\oplus$ følger, at $\mathbf{x} + \mathbf{y}$ netop vil have 1-taller på de pladser, hvor $\mathbf{x}$ og $\mathbf{y}$ er forskellige, det vil sige på $d(\mathbf{x}, \mathbf{y})$ pladser, og nuller på alle andre pladser. Men da vægten af et kodeord netop er antallet af 1-taller i kodeordet følger, at $w(\mathbf{x} + \mathbf{y}) = d(\mathbf{x}, \mathbf{y})$. □

Nu kan vi etablere vores genvej til bestemmelse af minimumsafstanden for en lineær kode.

**Sætning 4.7**
*For en lineær kode $\mathcal{C}$ gælder, at $d_{\min} = w_{\min}$.*

**Bevis**
Lad $\mathbf{x}$ og $\mathbf{y}$ være kodeord med $d(\mathbf{x}, \mathbf{y}) = d_{\min}$. Da gælder ifølge lemma 4.6, at

$$d_{\min} = d(\mathbf{x}, \mathbf{y}) = w(\mathbf{x} + \mathbf{y}) \geq w_{\min}.$$

På den anden side gælder, at hvis $w_{\min}$ er den mindste vægt i $\mathcal{C}$, så må der eksistere et kodeord $\mathbf{u} \in \mathcal{C}$ med $w(\mathbf{u}) = w_{\min}$. Altså, igen ifølge lemma 4.6,

$$w_{\min} = w(\mathbf{u}) = w(\mathbf{u} + \mathbf{0}) = d(\mathbf{u}, \mathbf{0}) \geq d_{\min}.$$

Ved at kombinere de to uligheder fås $d_{\min} = w_{\min}$. □

**Eksempel 4.8**
Koden $C$ fra eksempel 2.1 har $w_{\min} = 3$ (jævnfør eksempel 2.10). Det betyder, at $\mathcal{C}$ også har $d_{\min} = 3$ (se eventuelt også eksempel 2.14). ◇

## 4.3 En lille historie om Gauss

Vi er allerede blevet præsenteret for et par matematiske genier i løbet af dette undervisningsmateriale. Nu skal vi møde et nyt, og tilmed et af de største i matematikkens historie, den tyske matematiker Carl Friedrich Gauss (1777-1855).

Der er mange historier om Gauss og hans tidligt udviklede geni. Den af historierne vi skal høre udspillede sig da Gauss var ni år. Gauss gik i skole i Brunswick og i begyndelsen af skoleåret stillede Gauss' lærer, J.G. Büttner, en opgave der skulle holde de omkring 100 elever i klassen beskæftiget. Opgaven var at lægge tallene fra 1 til 100 sammen. Allerede inden Büttner var færdig med forklare opgaven til klassen havde Gauss skrevet tallet 5050 på sin tavle og afleveret denne på lærerens kateter. Nu kunne man tro at Gauss blot var usandsynligt hurtig til hovedregning, hvilket han sikkert også var, men årsagen var faktisk en anden. Gauss havde hurtigt indset, at svaret til spørgsmålet var 50 gange summen 101, hvilket kommer af at man kan danne 50 additioner med resultat 101:

$$100 + 1, \quad 99 + 2, \quad 98 + 3, \quad \ldots \quad 52 + 49, \quad 51 + 50,$$

og han havde med det samme udregnet dette resultat i hovedet. Imponeret af sin unge elev arrangerede Büttner, at Gauss fik specielle lærebøger og ekstraundervisning af Büttners assistent Martin Bartels (1769-1836), som selv senere blev professor i matematik i Rusland. Som resultat af denne undervisning blev Gauss snart sendt i gymnasium, hvor han mestrede det klassiske pensum.

Gauss' måde at løse opgaven ovenfor kan opskrives til en generel formel

$$g + (g-1) + (g-2) + \ldots + 1 = \frac{g(g+1)}{2}$$

og beviset for formlen bygger netop på denne paring af tal fra hver sin ende af talrækken som Gauss indså. Den formel vi i forbindelse med sammenligning af $m$ kodeord så i afsnit 4.2,

$$(m-1) + (m-2) + \ldots + 1 = \frac{m(m-1)}{2},$$

er blot en omskrivning af Gauss' formel, hvor vi sætter $g = m - 1$.

## 4.4 Lineær algebra og linearitetsbegrebet

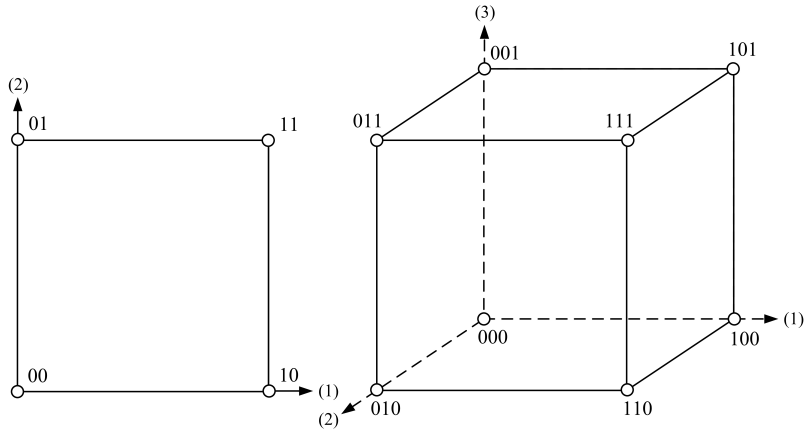
I afsnit 2.4 diskuterede vi de *rum*, som en mængde af punkter sammen med et generaliseret afstandsbegreb giver anledning til. For vores blokkoder er der som sagt tale om, at de $2^n$ talsæt sammen med Hamming-afstanden udgør et sådant (metrisk) rum. En form for billedliggørelse af sådanne rum så vi i afsnit 3.3, hvor vi så på en mere geometrisk tilgang til blokkoderne. I figur 3.1 i dette afsnit illustrerede vi talsæt ved symbolet $\circ$ og kodeord ved $\bullet$. Imidlertid tog vi det ikke så nøje, hvad længden af kodeordene var, det vil sige hvad $n$ var, vi illustrede blot situationen med kuglerne i henholdsvis 2 og 3 dimensioner for vilkårlige $n$. Størrelsen af $n$ er dog ikke uden betydning og i dette afsnit skal vi i højere grad koncentrere os om denne.

Hvis vi ser på mængden af talsæt $\{00, 01, 10, 11\}$ kan vi tænke på denne som placeret i 1. kvadrant af et koordinatsystem i planen. Talsættet 00 vil ligge i origo, 01 vil ligge én op af andenaksen, 10 vil ligge én ud af førsteaksen og 11 vil ligge i punktet $(1, 1)$. På lignende vis kan mængden af talsæt $\{000, 100, 010, 001, 011, 110, 101, 111\}$ placeres i 1. kvadrant af et 3 dimensionalt koordinatsystem; 000 vil ligge i origo, 100 vil ligge én ud af førsteaksen, og så videre. Faktisk vil der være tale om, at man kan tænke på talsættene som udgørende de otte hjørner af en 3-dimensional terning. Disse to situationer er illustreret på figur 4.1.

Men hvordan skal vi nu fortolke talsæt af længde $n \geq 4$? En geometrisk billedliggørelse af disse på samme måde som i figur 4.1 er ikke mulig, vi kan simpelthen ikke 'se' det for os. Fra en matematisk synsvinkel er dette imidlertid ikke så afgørende, vi kan nemlig sagtens regne med talsæt af længde $n \geq 4$ som punkter i $n$-dimensionale rum, selv om vi ikke kan forestille os disse rum. Derfor giver det altså mening, for vilkårlige $n$, at tænke på $n$-dimensionale terninger i 1. kvadrant af et koordinatsystem med $n$ akser, hvor alle $n$ akser 'står vinkelret' på hinanden.

Historisk set var denne generalisation til $n$-dimensionale rum længe undervejs. Matematikhistorikeren Victor Katz siger:

> Geometri som det blev studeret frem til midten af det nittende århundrede beskæftigede sig ikke med objekter af dimensioner større end tre. Men med den tiltagende brug af analytiske og




**Figur 4.1** Binære talsæt af længde 2 afbilledet som punkter i planen og binære talsæt af længde 3 afbilledet som punkter i rummet.

> algebraiske metoder blev det klart, at der for mange geometriske ideer ikke var nogen speciel grund til at begrænse antallet af dimensioner til det antal som kunne forstås fysisk. Altså generaliserede adskillige matematikere deres formler og sætninger til $n$ dimensioner, hvor $n$ kunne være et vilkårligt positivt heltal. (Katz; 1998, side 767, oversat fra engelsk)

Det første detaljerede studie af $n$-dimensionale rum i en geometrisk sammenhæng blev udført af den tyske matematiker Hermann Günther Grassmann (1809-1877) i 1844 i hans værk *Die Ausdehnungslehre*. *Ausdehnung* er tysk for *udstrækning*, så det er altså *læren om udstrækning*, men siden det er dimensionerne som Grassmann er ude på at 'udstrække' oversættes titlen måske bedre til *dimensionalitetslære*. Grassmann var selvlært matematiker og derfor fuldstændig ukendt for andre matematikere på sin tid. Af den årsag forblev hans arbejde også ukendt og det til trods for at han i 1862 udgav en fuldstændig revideret version af dette som levede op til datidens krav om matematisk stringens, notation og lignende. Grassmann definerer i sit arbejde en væsentlig del af de begreber der i dag spiller en så betydelig rolle inden for den matematiske disciplin kendt som *lineær algebra*. Lineær algebra beskæftiger sig med studierne af additivitet og proportionalitet og *linearitetsbegrebet* er det begreb der knytter netop disse to egenskaber sammen. (I definition 4.3 udgør betingelse *i* additiviteten og betingelse *ii* proportionaliteten.) Fordi Grassmanns arbejde var ukendt for de fleste kom dette historisk set ikke til at betyde noget videre i udviklingen af den lineære algebra. Det gjorde arbejder af for eksempel den irske matematiker William Rowan Hamilton (1805-1865) og den engelske matematiker Arthur Cayley (1821-1895) derimod. Den lineære algebra og herunder teorien for $n$-dimensionale rum, også kaldet vektorrum, blev stort set udviklet uafhængigt af Grassmann, omend mange af de resultater man fandt frem til var identiske med Grassmanns.

Omkring 1920 gav Grassmanns *Ausdehnungslehre* dog inspiration til andre matematikere som videreudviklede områder af den lineære algebra. Den fulde accept og forståelse af dybden i Grassmanns arbejde kom først endnu senere i løbet af det tyvende århundrede.

Selve ideen eller ønsket om den teori som Grassmann udviklede kan imidlertid spores endnu længere tilbage. Allerede i 1679 skrev Leibniz i et brev til den hollandske matematiker Christiaan Huygens (1629-1695):

> Der findes en måde til at bringe algebra lige så langt videre i forhold til hvad Viète og Descartes har efterladt os som Viète og Descartes førte den videre fra oldtiden. [...] Vi behøver en analyse som er tydelig geometrisk eller lineær... (Leibniz, 1679, oversat fra engelsk i Fearnley-Sander (1979, side 809))

François Viète (1540-1603) og René Descartes (1596-1650) som Leibniz omtaler her var tidligere matematikere som havde gjort store fremskridt inden for matematikken. Leibniz' korrespondance med Huygens blev dog ikke offentliggjort førend i 1833 og det er uvist, hvorvidt Grassmann var klar over, at han udviklede den teori som Leibniz i 1679 drømte om.

Ideen om linearitet, altså det begreb der knytter additivitet og proportionalitet sammen, kan formentligt spores endnu længere tilbage end studiet af $n$-dimensionale rum. Eksempelvis omtaler matematikhistorikeren Jean-Luc Dorier i denne forbindelse to (uafhængige) afhandlinger af de to schweiziske matematikere, Leonhard Euler (1707-1783) og Gabriel Cramer (1704-1752), dateret 1750. Vi skal imidlertid ikke gå i dybden med dette, men i stedet se på Hammings $(7,4)$-kode – en kode som opfylder kravet om linearitet.

## 4.5 $(7,4)$-Hamming-koden og dens afkodning

Som vi så i afsnit 4.2 er lineære koder attraktive, eksempelvis fordi de er nemme at bestemme minimumsafstanden for. Vi skal nu se et eksempel på, hvorledes man kan konstruere en lineær kode[1]. Vi skal konstruere en kode $\mathcal{H}$ med $2^4 = 16$ kodeord og bloklængde $n = 7$. Det viser sig at være snedigt at konstruere koden på en sådan vis, at et talsæt $(x_1, x_2, x_3, x_4, x_5, x_6, x_7)$ er et kodeord, hvis

$$\begin{aligned} x_5 &= 0x_1 \oplus 1x_2 \oplus 1x_3 \oplus 1x_4 \\ x_6 &= 1x_1 \oplus 0x_2 \oplus 1x_3 \oplus 1x_4 \\ x_7 &= 1x_1 \oplus 1x_2 \oplus 0x_3 \oplus 1x_4 \end{aligned} \tag{4.1}$$

Koden består altså af følgende seksten kodeord, hvor $x_1$, $x_2$, $x_3$ og $x_4$ antager alle kombinationer af 0 og 1:

$$\mathcal{H} = \left\{ \begin{array}{cccc} 0000000, & 1000011, & 0100101, & 0010110, \\ 0001111, & 1100110, & 1010101, & 1001100, \\ 0110011, & 0101010, & 0011001, & 1110000, \\ 0111100, & 1011010, & 1101001, & 1111111 \end{array} \right\}$$

[1] Afsnit 4.5 er i høj grad baseret på fremstillingen i (Jensen & Høholdt; 1993, side 20-26).

Af selve fremgangsmåden for konstruktionen af koden ses det, at man selv er i stand til at disponere over de første fire cifre, $x_1, x_2, x_3, x_4$, i et kodeord. Disse fire pladser udgøres derfor af dataordet og kaldes *informationscifre*. De sidste tre, altså $x_5, x_6, x_7$, som jo er fastlagt på baggrund af informationscifrene, kaldes *kontrolcifre*. Informationscifrene og kontrolcifrene udgør altså tilsammen kodeordet. Koden $\mathcal{H}$ er derfor en $(7,4)$-kode og rent faktisk er det den $(7,4)$-Hamming-kode som Shannon brugte i sin 1948-artikel.

Ved hjælp af sætning 4.7 er vi på en nem måde i stand til at finde minimumsafstanden for koden $\mathcal{H}$, nemlig ved at bestemme den mindste vægt fraregnet vægten af nul-kodeordet. Denne er 3, da det mindste antal 1-taller i et kodeord forskelligt fra nul-kodeordet er 3. (Uden sætning 4.7 havde vi været nødsaget til at foretage i alt 120 parvise sammenligninger af kodeordne i $\mathcal{H}$ for at bestemme minimumsafstanden. Hvorfor netop 120?) Mere generelt er $(7,4)$-Hamming-koden altså en $(7,4,3)$-blokkode. Ifølge sætning 3.6 og sætning 3.7 er $(7,4)$-Hamming-koden en 2-fejldetekterende kode og en 1-fejlkorrigerende kode.

Som sagt foretog Golay, på baggrund af Shannons 1948-artikel, generaliseringen af Hammings $(7,4)$-kode. Vi skal her antyde nogle af elementerne i en sådan generalisering og derefter give en beskrivelse af Hammings afkodningsprocedure – en procedure som også blev eksemplificeret i Shannons artikel.

Med udgangspunkt i konstruktionen af $(7,4)$-koden er vi altså interesserede i at kunne indkode $2^k$ dataord, $(x_1, x_2, \ldots, x_k)$, for således at konstruere en kode med bloklængde $n$ ved at tilføje kontrolcifrene $x_{k+1}, x_{k+2}, \ldots, x_n$. Dette gøres efter følgende regel:

$$\begin{aligned} x_{k+1} &= a_{11}x_1 \oplus a_{12}x_2 \oplus \ldots \oplus a_{1k}x_k \\ x_{k+2} &= a_{21}x_1 \oplus a_{22}x_2 \oplus \ldots \oplus a_{2k}x_k \\ &\vdots \\ x_n &= a_{n-k\,1}x_1 \oplus a_{n-k\,2}x_2 \oplus \ldots \oplus a_{n-k\,k}x_k, \end{aligned} \tag{4.2}$$

hvor $a_{ij}$, for $i = 1, \ldots, n-k$ og $j = k, \ldots, n$, er enten 0 eller 1. Problemet består nu selvfølgelig i, hvorledes man kan vælge $a_{ij}$'erne på den mest fordelagtige vis, det vil sige således at koden bliver i stand til at rette flest mulige fejl. Dette skal vi ikke gå i dybden med her, istedet skal vi koncentrere os om Hammings afkodningsprocedure.

Det centrale begreb i afkodningsproceduren er det der kaldes *syndromet*. Syndromet er, ligesom mange af de andre begreber vi indtil nu har stiftet bekendtskab med, en organiseret måde, hvorpå vi ved manipulation kan anskueliggøre endnu et aspekt af koderne. Til definition af dette begreb bemærker vi først, at den kode der er beskrevet i (4.2) også kan beskrives ved, at $\mathbf{x} = (x_1, x_2, \ldots, x_n)$ er et kodeord i $(n,k)$-koden hvis og kun hvis der gælder:

$$0 = x_{k+1} \oplus a_{11}x_1 \oplus a_{12}x_2 \oplus \ldots \oplus a_{1k}x_k$$

$$\begin{aligned} 0 &= x_{k+2} \oplus a_{21}x_1 \oplus a_{22}x_2 \oplus \ldots \oplus a_{2k}x_k \\ &\vdots \\ 0 &= x_n \oplus a_{n-k\,1}x_1 \oplus a_{n-k\,2}x_2 \oplus \ldots \oplus a_{n-k\,k}x_k. \end{aligned} \tag{4.3}$$

**Definition 4.9: Syndromet**
*Lad* $\mathbf{z} = (z_1, z_2, \ldots, z_n)$ *være et vilkårligt talsæt (eller ord) af længde* $n$. Syndromet *er da defineret som* $\mathrm{s}(\mathbf{z}) = (s_1, s_2, \ldots, s_{n-k})$, *hvor*

$$\begin{aligned} s_1 &= z_{k+1} \oplus a_{11}z_1 \oplus a_{12}z_2 \oplus \ldots \oplus a_{1k}z_k \\ s_2 &= z_{k+2} \oplus a_{21}z_1 \oplus a_{22}z_2 \oplus \ldots \oplus a_{2k}z_k \\ &\vdots \\ s_{n-k} &= z_n \oplus a_{n-k\,1}z_1 \oplus a_{n-k\,2}z_2 \oplus \ldots \oplus a_{n-k\,k}z_k. \end{aligned} \tag{4.4}$$

**Eksempel 4.10**
I tilfældet med Hammings $(7, 4)$-kode, se (4.1), kan vi nu for et vilkårligt ord $\mathbf{z}$ af længde 7, $(z_1, z_2, \ldots, z_7)$, beregne syndromet $\mathrm{s}(\mathbf{z}) = (s_1, s_2, s_3)$ givet ved

$$\begin{aligned} s_1 &= x_5 \oplus 0x_1 \oplus 1x_2 \oplus 1x_3 \oplus 1x_4 \\ s_2 &= x_6 \oplus 1x_1 \oplus 0x_2 \oplus 1x_3 \oplus 1x_4 \\ s_3 &= x_7 \oplus 1x_1 \oplus 1x_2 \oplus 0x_3 \oplus 1x_4. \end{aligned} \tag{4.5}$$

Eksempelvis gælder der, at

$$\begin{aligned} \mathrm{s}(1110111) &= (111) \\ \mathrm{s}(0110000) &= (011) \\ \mathrm{s}(1111111) &= (000). \end{aligned}$$

◇

Syndromet har visse egenskaber som er væsentlige for afkodningsproceduren. Vi sammenfatter disse i en sætning.

**Sætning 4.11**
*Lad der være givet en lineær kode* $\mathcal{C}$ *med bloklængde* $n$, *da gælder:*

*i.* $\mathrm{s}(\mathbf{x} + \mathbf{y}) = s(\mathbf{x}) + s(\mathbf{y})$, *hvor* $\mathbf{x}$ *og* $\mathbf{y}$ *er vilkårlige ord af længde* $n$.
*ii.* $\mathrm{s}(\mathbf{x}) = 0$ *hvis og kun hvis* $\mathbf{x}$ *er et kodeord i* $\mathcal{C}$.
*iii.* $\mathrm{s}(\mathbf{x}) = \mathrm{s}(\mathbf{y})$ *hvis og kun hvis* $\mathbf{x} + \mathbf{y}$ *er et kodeord i* $\mathcal{C}$.

**Bevis**
Egenskab *i* fås af de ligninger som bestemmer syndromet (jævnfør 4.4). Beviset er i og for sig forholdsvis ligetil men kræver en del plads, hvorfor det er henlagt til opgave 45.

Egenskab *ii* følger umiddelbart af definitionerne af henholdsvis koden og syndromet: Hvis $\mathbf{x}$ er et kodeord, så er alle ligninger i (4.3) lig 0, hvorfor $\mathrm{s}(\mathbf{x}) = \mathbf{0}$. Hvis derimod $\mathrm{s}(\mathbf{x}) = \mathbf{0}$, det vil sige alle ligninger i (4.4) er lig 0, så følger det jo af (4.3) at $\mathbf{x}$ netop er et kodeord.

Egenskab *iii* vises ved at benytte de to foregående egenskaber: Hvis $\mathbf{x} + \mathbf{y}$ er et kodeord gælder der ifølge egenskab *ii*, at $\mathrm{s}(\mathbf{x} + \mathbf{y}) = \mathbf{0}$. Ifølge egenskab *i* har vi derfor, at $\mathrm{s}(\mathbf{x}) + \mathrm{s}(\mathbf{y}) = \mathbf{0}$, men det kan jo kun lade sig gøre, hvis $\mathrm{s}(\mathbf{x}) = \mathrm{s}(\mathbf{y})$. Bemærk, at dette argument så at sige også gælder baglæns, altså

$$\mathrm{s}(\mathbf{x}) = \mathrm{s}(\mathbf{y}) \Leftrightarrow \mathrm{s}(\mathbf{x}) + \mathrm{s}(\mathbf{y}) = \mathbf{0} \Leftrightarrow \mathrm{s}(\mathbf{x} + \mathbf{y}) = \mathbf{0},$$

hvilket faktisk viser begge veje i 'hvis og kun hvis'-sætningen. □

Endvidere følger af sætning 4.11, at hvis $\mathbf{u}$ er et vilkårligt ord og $\mathbf{x}$ er et kodeord, så er

$$\mathrm{s}(\mathbf{u} + \mathbf{x}) = \mathrm{s}(\mathbf{u}) + \mathrm{s}(\mathbf{x}) = \mathrm{s}(\mathbf{u}) + \mathbf{0} = \mathrm{s}(\mathbf{u}).$$

Vi er nu i stand til at præsentere afkodningsproceduren, der også er kendt som *syndrom-afkodning*. På baggrund af et modtaget ord $\mathbf{v}$ udregnes $\mathrm{s}(\mathbf{v})$. Lad $\mathbf{f}$ være et ord med mindst vægt der har samme syndrom, $\mathrm{s}(\mathbf{f}) = \mathrm{s}(\mathbf{v})$. ($\mathbf{f}$ må her godt være lig $\mathbf{0}$, hvorfor $w(\mathbf{f})$ i dette tilfælde godt kan være 0.) Ifølge sætning 4.11 er $\mathbf{v} + \mathbf{f}$ et kodeord og det viser sig, at det tilmed er et kodeord der ligger tættest på $\mathbf{v}$. Dette indses på følgende vis: Lad $\mathbf{x}$ være et vilkårligt kodeord. Der gælder, at

$$d(\mathbf{v}, \mathbf{x}) = d(\mathbf{v} + \mathbf{x}, \mathbf{0}) = w(\mathbf{v} + \mathbf{x})$$

(jævnfør eventuelt opgave 23), hvor

$$\mathrm{s}(\mathbf{v} + \mathbf{x}) = \mathrm{s}(\mathbf{v}) = \mathrm{s}(\mathbf{f}),$$

da $\mathbf{x}$ jo er et kodeord. Så på grund af valget af $\mathbf{f}$ gælder, at

$$d(\mathbf{v}, \mathbf{x}) \geq w(\mathbf{f}) = d(\mathbf{f}, \mathbf{0}) = d(\mathbf{v} + \mathbf{f}, \mathbf{v}).$$

Altså er $\mathbf{v} + \mathbf{f}$ et kodeord der ligger tættest på $\mathbf{v}$.

Ligesom 'nærmeste-nabo'-afkodning afkoder syndrom-afkodning altså også et modtaget ord til dets nærmeste nabo. Måden det sker på er blot mere effektiv. Ideen er, at man for hvert af de $2^{n-k}$ syndromer (der er lige så mange syndromer som der er kontrolcifre) identificerer det ord med mindst vægt som giver anledning til det pågældende syndrom. Et sådant ord kaldes for *klasseføreren* hørende til syndromet – ordet $\mathbf{f}$ i gennemgangen ovenfor var en sådan klassefører. Klasseførerne opstilles i en liste arrangeret efter de tilhørende syndromer. Selve afkodningen sker ved, at man for et modtaget ord $\mathbf{v}$ udregner syndromet, $\mathrm{s}(\mathbf{v})$, lokaliserer dette i listen og adderer den tilhørende klassefører til $\mathbf{v}$ for således at få det oprindeligt afsendte kodeord. Lad os se et eksempel.

**Eksempel 4.12**
I tilfældet med $(7,4)$-Hamming-koden kan vi med udgangspunkt i (4.5) opskrive følgende sammenhænge mellem de $2^3$ syndromer og de ord med mindst vægt (klasseførerne) som giver disse:

$$\begin{aligned}(000) &= \mathrm{s}(0000000)\\ (001) &= \mathrm{s}(0000001)\\ (010) &= \mathrm{s}(0000010)\\ (100) &= \mathrm{s}(0000100)\\ (110) &= \mathrm{s}(0010000)\\ (011) &= \mathrm{s}(1000000)\\ (101) &= \mathrm{s}(0100000)\\ (111) &= \mathrm{s}(0001000).\end{aligned}$$

Modtager vi eksempelvis ordet $\mathbf{v} = (0011110)$ fås $\mathrm{s}(\mathbf{v}) = (111)$ og vi afkoder derfor ved $(0011110) + (0001000) = (0010110)$, hvilket er vores kodeord. På lignende vis vil vi for eksempel kunne korrigere fejlen i det modtagne ord $\mathbf{v} = (1100111)$ og få kodeordet (1100110). Regn selv efter! ◇

$(7,4)$-Hamming-koden, og Hamming-koderne generelt, er altså eksempler på de føromtalte 'rimelige' koder. Det vil sige koder som har et fornuftigt forhold mellem antallet af informationscifre og antallet af kontrolcifre, eller med andre ord en fornuftig informationsrate, og samtidig ikke er alt for komplicerede at ind- og afkode. Specielt afkodningen er rimelig ukompliceret forstået på den måde, at det $\mathbf{f}$ der indgår i proceduren er valgt på forhånd og ikke ændrer sig.

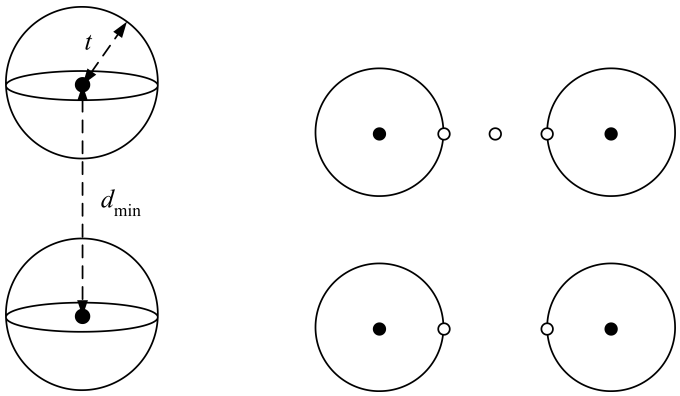
## 4.6 Perfekte koder

For en kode med minimumsafstand $2t+1$ gælder som tidligere vist (se sætning 3.10), at Hamming-kugler med radius $t$ placeret med centrum i kodens kodeord ikke vil overlappe hinanden (se illustrationen til venstre i figur 4.2). Denne observation giver anledning til definitionen af et nyt begreb, nemlig *pakningsradius*.

**Definition 4.13**
*Lad $\mathcal{C}$ være en kode med minimumsafstand $d_{\min}$. Kodens* pakningsradius*, $p(\mathcal{C})$, er det største positive heltal for hvilket mængden af kugler med radius $p(\mathcal{C})$ centreret i $\mathcal{C}$'s kodeord er ikke-overlappende.*

(Jævnfør definitionen af en Hamming-kugle, definition 3.9, er der altså tale om at pakningsradiusen for en given kode er en ganske bestemt radius $r$.)

Værdien af pakningsradiusen afhænger af minimumsafstanden $d_{\min}$ af koden. Hvis $d_{\min}$ er et *lige* tal, lad os sige $d_{\min} = 2t+2$, øger vi radiusen




**Figur 4.2** Til venstre: Geometrisk fremstilling af minimumsafstanden $d_{\min}$ mellem rumlige kugler. Øverst til højre: Radius øget indtil lige før kuglerne *rører* hinanden. Nederst til højre: Radius øget indtil lige før kuglerne *overlapper* hinanden.

af kuglerne indtil lige før de to kugler *rører* hinanden. Dette er illustreret øverst til højre på figur 4.2. I dette tilfælde er pakningsradius givet ved

$$p(\mathcal{C}) = t = \frac{d_{\min} - 2}{2}.$$

Hvis $d_{\min}$ derimod er et *ulige* tal, $d_{\min} = 2t+1$, øger vi radiusen af kuglerne indtil lige før de to kugler *overlapper* hinanden. Dette er illustreret nederst til højre på figur 4.2. I dette tilfælde er pakningsradius givet ved

$$p(\mathcal{C}) = t = \frac{d_{\min} - 1}{2}.$$

Ved at indføre en matematisk funktion kaldet *heltalsdelen* kan vi sammenfatte ovenstående i en sætning.

**Definition 4.14**
Heltalsdelen *af et reelt tal $a$ er det største heltal mindre end eller lig $a$. Heltalsdelen af $a$ skrives $\lfloor a \rfloor$.*

**Eksempel 4.15**

$$\lfloor 3,5 \rfloor = 3 \qquad \lfloor 9,999999 \rfloor = 9 \qquad \left\lfloor \frac{1}{2} \right\rfloor = 0 \quad \left\lfloor \frac{4}{3} \right\rfloor = 1 \quad \left\lfloor 8\frac{1}{8} \right\rfloor = 8$$

◇

**Sætning 4.16**
*Pakningsradiusen for en kode med minimumsafstand $d_{\min}$ er givet ved*

$$p(\mathcal{C}) = \left\lfloor \frac{d_{\min} - 1}{2} \right\rfloor.$$

**Bevis**

$$\frac{d_{\min}-2}{2} = \left\lfloor \frac{d_{\min}-1}{2} \right\rfloor \qquad \text{og} \qquad \frac{d_{\min}-1}{2} = \left\lfloor \frac{d_{\min}-1}{2} \right\rfloor .$$

□

Når vi taler om pakningsradius $p(\mathcal{C})$ vil vi kalde kuglerne for *pakningskugler*. Disse er i bund og grund de samme som Hamming-kugler blot med den forskel, at nu er radiussen altså fastlagt og kan ikke vælges frit. Kuglerne centreret i kodeordene på figur 4.2 er sådanne pakningskugler.

Nogle ganske få men vigtige koder besidder en egenskab som er af så stor betydning, at koderne tildeles en særlig betegnelse.

**Definition 4.17: En perfekt kode**
*Lad $\mathcal{C} = \{\mathbf{x}_1, \ldots, \mathbf{x}_m\}$ være en binær kode med bloklængde $n$ og pakningsradius $p(\mathcal{C})$. Lad $B$ være mængden af alle binære talsæt af længde $n$. Koden $\mathcal{C}$ kaldes* perfekt*, hvis ethvert element $\mathbf{b} \in B$ er indeholdt i en pakningskugle, det vil sige*

$$B \subseteq S(\mathbf{x}_1, p(\mathcal{C})) \cup \ldots \cup S(\mathbf{x}_m, p(\mathcal{C})).$$

Da kuglerne ikke overlapper, er der i definition 4.17 i virkeligheden tale om, at foreningen af pakningskuglerne på højresiden i udtrykket ovenfor giver hele mængden $B$, det vil sige

$$B = S(\mathbf{x}_1, p(\mathcal{C})) \cup \ldots \cup S(\mathbf{x}_m, p(\mathcal{C})).$$

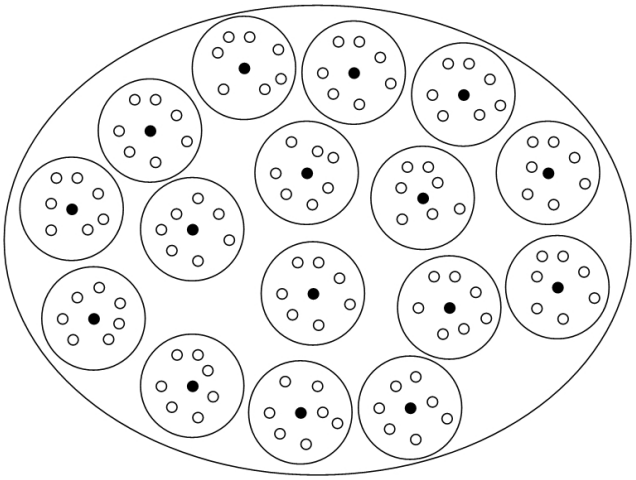
Der er altså tale om, at pakningskuglerne så at sige 'fylder rummet ud' og det er formentlig herfra at de perfekte koder har fået deres navn.

$(7,4)$-Hamming-koden er et eksempel på en sådan perfekt kode. I dette tilfælde består mængden $B$ af $2^7 = 128$ talsæt af længde 7, hvoraf kun de seksten er kodeord. Vi kan nu danne seksten ikke-overlappende pakningskugler med centrum i kodeordene og pakningsradius $p(\mathcal{C}) = 1$. Det betyder, at hver kugle med pakningsradius foruden kodeordet vil indeholde syv talsæt af længde 7, hver med Hamming-afstand 1 til kodeordet. For nul-kodeordet vil vi eksempelvis have

$$S(\mathbf{0}, 1) = \{1000000, 0100000, \ldots, 0000001\}.$$

På figur 4.3 er 'geometrien' for $(7,4)$-Hamming-koden illustreret.

Den såkaldte repetitionskode er et andet eksempel på en perfekt kode. En binær *repetitionskode* kan som udgangspunkt kun indkode to dataord, nemlig 0 og 1. Indkodningen foregår ved at dataordet gentages $n$ gange. Herved får man en $(n,1)$-kode, med minimumsafstand $n$.



**Figur 4.3** En illustration af en perfekt kode – nærmere bestemt den binære $(7,4)$-Hamming-kode. Antallet af kodeord for denne er $2^4 = 16$ og antallet af binære talsæt af længde 7 er $2^7 = 128$. Alle disse talsæt er indeholdt i en pakningskugle. Hver pakningskugle indeholder $128/16 = 8$ talsæt, hvoraf det ene er et kodeord.

**Eksempel 4.18 (Repetitionskode)**
For $n = 5$ fås en kode, som giver følgende indkodning:

$$\begin{aligned} 0 &\mapsto 00000 \\ 1 &\mapsto 11111 \end{aligned}$$

Repetitionskoden har i modsætning til paritetscheckkoden, som vi så tidligere, fantastiske fejlkorrigerende egenskaber. Der kan nemlig ifølge sætning 4.16 korrigeres $\lfloor (n-1)/2 \rfloor$ fejl. Til gengæld har repetitionskoden en forfærdelig lav informationsrate, nemlig $1/n$. ◇

For en repetitionskode gælder, at hvis minimumsafstanden er ulige, er koden perfekt. I eksempel 4.18 ovenfor kan der placeres kugler med radius 2 omkring de to kodeord, således at kuglerne er ikke-overlappende samtidigt med, at de dækker hele mængden $B$. Repetitionskoder af ulige længde, koder bestående af kun ét kodeord samt koder bestående af alle talsæt i $B$ kendes som *trivielle perfekte koder* (jævnfør side 26). Fælles for repetionskoder af ulige længde, koder bestående af alle talsæt i $B$ og $(7,4)$-Hamming-koden er, at minimumsafstanden altid er et ulige tal. Dette er ingen tilfældighed, der gælder nemlig følgende sætning:

**Sætning 4.19**
*En perfekt kode har altid ulige minimumsafstand.*

**Bevis**
Hvis vi kan vise, at en kode med lige minimumsafstand ikke kan være perfekt, må der nødvendigvis gælde, at en perfekt kode har ulige minimumsafstand (men jo bestemt ikke det omvendte).

Vi antager, at en kode $\mathcal{C}$ har lige minimumsafstand $2t$. Lad nu $\mathbf{x}$ og $\mathbf{y}$ være to kodeord i $\mathcal{C}$ med $d(\mathbf{x}, \mathbf{y}) = 2t$ og antag endvidere, at det er på de første $2t$ pladser $\mathbf{x}$ og $\mathbf{y}$ er forskellige. Lad nu $\mathbf{v}$ være ordet som er lig $\mathbf{x}$ på de første $t$ pladser, lig $\mathbf{y}$ på de næste $t$ pladser og lig både $\mathbf{x}$ og $\mathbf{y}$ på de resterende pladser. Vi har da, at

$$d(\mathbf{x}, \mathbf{v}) = d(\mathbf{y}, \mathbf{v}) = t.$$

Imidlertid gælder der for koden $\mathcal{C}$, at

$$p(\mathcal{C}) = \left\lfloor \frac{2t-1}{2} \right\rfloor = t - 1 < t,$$

hvorfor ingen pakningskugle vil indeholde $\mathbf{v}$, hvilket er det samme som at sige, at $\mathcal{C}$ ikke er perfekt. □

Perfekte koder forekommer af flere årsager sjældent i praktiske anvendelser: For det første er kodernes parametre $(n, m, k)$ helt fastlåste, hvorfor de ikke kan tilpasses særlige krav i anvendelserne. For det andet har koderne den svaghed, at hvis der opstår for mange fejl ved transmissionen vil afkodningen med garanti føre til en afkodningsfejl.

## 4.7 Hamming-koder og Golay-koder

Listen af ikke-trivielle perfekte koder er kort. Den omfatter blot familien af Hamming-koder samt de to perfekte Golay-koder (af de koder man kender til i hvert fald – der kan selvfølgelig i princippet findes andre).

Familien af binære Hamming-koder benævnes undertiden $\mathcal{H}_2(h)$, hvor $h$ er antallet af kontrolcifre i kodens kodeord, altså $n - k$. Det kan vises, at for hvert positive heltal $h$ er den binære Hamming-kode $\mathcal{H}_2(h)$ en $(n, k, d_{\min})$-kode med parametre

$$n = 2^h - 1, \quad k = 2^h - h - 1, \quad d_{\min} = 3, \quad m = 2^{n-h},$$

hvor $m$ ligesom tidligere er antallet af kodeord i koden. Bemærk, at alle Hamming-koder, underordnet værdien af parametrene, altid vil have minimumsafstand 3 – et resultat som vi ikke skal vise her. En anden ting der er værd at bemærke sig er, at blot en lille stigning af $h$-værdien øger antallet af kodeord, $m$, i Hamming-koden markant. En $h$-værdi på 3 giver os den velkendte $(7, 4)$-Hamming-kode med 16 kodeord. Hamming-koden med $h = 4$ består derimod af 2.048 kodeord. Situationen er illustreret i tabel 4.2 for $h = 2, 3, 4, 5, 6$.

Et nærliggende spørgsmål er nu hvad man dog i alverden skal bruge for eksempel 2.048 forskellige kodeord til, når vi fra ASCII-koden ved, at vi oftest kun er interesseret i at skrive 128 forskellige tegn. Ja, forestil dig, at

| $h$ | $n = 2^h - 1$ | $k = n - h$ | $m = 2^{n-h}$ |
|---|---|---|---|
| 2 | 3 | 1 | 2 |
| 3 | 7 | 4 | 16 |
| 4 | 15 | 11 | 2.048 |
| 5 | 31 | 26 | 67.108.864 |
| 6 | 63 | 57 | 144.115.188.075.855.872 |

**Tabel 4.2** Parameterværdier for binære Hamming-koder ordnet efter forskellige værdier af $h$.

Zorro i stedet for at sende beskeden $Z$ til sergent Garcia ønskede at sende et farvebillede af sig eget (det vil sige Guy Williams) smørede grin. I så tilfælde vil 2.048 kodeord til at dække forskellige farvenuancer i billedet ikke være for mange. Vi skal berøre brug af fejlkorrigerende koder i forbindelse med transmission af digitale billeder igen i afsnit 4.9.

Golay-koderne består af blot fire koder, hvoraf kun de to er perfekte; en binær og en ternær kode. Med ternær forstås at kodens alfabet er givet ved $A = \{0, 1, 2\}$. Den binære kode $\mathcal{G}_{23}$ er en $(23, 12, 7)$-blokkode, og den ternære kode $\mathcal{G}_{11}$ er en $(11, 6, 5)$-blokkode. Den perfekte binære Golay-kode har altså $m = 2^{12} = 4.096$ kodeord og den ternære har $m = 3^6 = 729$, hvorfor vi skal afstå fra at opskrive koderne her. Golay benyttede i sin 1949-artikel en kompakt opskrivningsform for sine koder, hvilket gjorde det muligt for andre matematikere at generere koderne selv. En beskrivelse af denne opskrivningsform her vil imidlertid være for omfattende. I stedet skal vi se på nogle af de matematiske resultater der er vist for henholdsvis Golay- og Hamming-koderne.

Som nævnt i begyndelsen af dette afsnit er de eneste ikke-trivielle perfekte koder formentlig netop Hamming- og Golay-koderne. Man har vist, at for alfabeter $A = \{a_1, a_2, a_3, \ldots, a_p\}$ af symboler, hvor $p$ er et primtal, vil alle ikke-trivielle perfekte koder have samme parameterværdier som enten en Hamming- eller en Golay-kode. For en kode med samme parameterværdier som en Golay-kode gælder, at denne i bund og grund er den samme kode som Golay-koden. Og for en lineær kode med samme parameterværdier som en Hamming-kode, gælder at denne i bund og grund er den samme som Hamming-koden. Man siger, at koderne er *ækvivalente*. Over et vilkårligt alfabet er det endvidere vist, at den eneste ikke-trivielle $t$-fejlkorrigerende perfekte kode med $t \geq 3$ er den binære Golay-kode $\mathcal{G}_{23}$. Resultatet er vigtigt, fordi det overflødiggør yderligere søgning efter perfekte koder der kan rette mere end tre fejl.

Det er dog værd at bemærke, at der stadig findes ubesvarede spørgsmål med hensyn til Hamming- og Golay-koderne. Eksempelvis vides det ikke, hvor mange ikke-ækvivalente ikke-lineære perfekte koder med Hamming-parametre der findes over alfabeter af primtalsstørrelse (man kender eksempler på få sådanne koder). Og mere generelt vides det heller ikke om der findes perfekte 2-fejlkorrigerende koder over alfabeter som ikke er af primtalsstørrelse (selvom det formodes ikke at være tilfældet).

## 4.8 Et matematikhistorisk spørgsmål

Hvis man er interesseret i videnskabshistorie, herunder matematikhistorie, så kan det være af en vis betydning at forstå hvem der gjorde hvad først. Ikke mindst på grund af den videre udvikling af et videnskabeligt område kan det være vigtigt at vide hvem der var ophavsmanden (eller -kvinden) til et givet begreb eller en given teori, hvad der inspirerede til udviklingen af begrebet eller teorien samt i hvilket omfang ophavsmanden kommunikerede med andre forskere om emnet.

Diskussionen om hvem der er den rigtige ophavsmand til familien af Hamming-koderne har været et diskuteret emne inden for kodningsteorien (og dens historie) siden 1950. Debatten går på hvorvidt Hamming kendte til generaliseringen af sin $(7, 4)$-kode, da han gav denne til Shannon i 1947-48 eller i modsat fald, om Golay var den første der opdagede dette i sin 1949-artikel. En udbredt antagelse er, at Hamming formentlig var udmærket klar over generaliserbarheden af $(7, 4)$-koden. Imidlertid var han, som vi hørte om tidligere, da han arbejdede ved den private virksomhed Bell Labs, nødt til at vente med at offentliggøre familien af Hamming-koder indtil sagen angående patentering var faldet på plads. Hamming siger selv om dette:

> Omtrent samtidigt med at informationsteori blev skabt, og på omtrent samme sted, blev kodningsteori også skabt. Den grundlæggende artikel blev imidlertid forsinket af patentkrav indtil april 1950, hvor den også dukkede op i *Bell System Technical Journal* [...] (Hamming; 1980, side 3, oversat fra engelsk)

At Golay foretog generalisering af $(7, 4)$-Hamming-koden er der ingen tvivl om, for eksempel fortæller en anden kodningsteoretiker ved navn Massey om Golays 1949-arbejde:

> Golay fortalte mig engang, at det eneste tidligere arbejde om kodning som han på det tidspunkt var vidende om var to-paragrafs beskrivelsen af $(7, 4)$-Hamming-koden som optræder i Shannons 1948-artikel. (Massey; 1990, side 2, oversat fra engelsk)

Diskussionen om generaliseringen af Hamming-koderne udmønter sig så eksempelvis i, hvem disse skal opkaldes efter. Og der er på dette punkt delte meninger, hvilket fremgår klart af følgende citat fra endnu en kodningsteoretiker, denne gang Berlekamp:

> Mange lærde personer fastholder at æren for forskning udelukkende skal gives til den første forfatter som publicerer samt at en uafhængig løsning bør modtage anerkendelse hvis og kun hvis modtagelsesdatoen for manuskriptet går forud for publikationen af den første artikel som forekommer. Selv om dette lyder fornuftigt kan denne simple regel være svær at anvende i praksis. Tag nu for eksempel Hamming-koderne. Den første publicerede reference til dette emne var 1948-artiklen af Shannon, som inkluderede $(7, 4)$-Hamming-koden som et eksempel, komplet med reference til Hamming. Efter at have læst Shannons artikel begyndte Golay at arbejde på fejlkorrigerende koder og opnåede nogle imponerende nye resultater, som for eksempel $(23, 12)$-Golay-koden,

> samt den forholdsvis ukomplicerede generalisering af Hammings $(7, 4)$-kode til alle de andre Hamming-koder. Begge blev publiceret før Hammings 1950-artikel. På grund af dette har visse lærde purister[2] bestemt, at det kun er $(7, 4)$-koden som tilhører Hamming. De har refereret til de andre enkelt-fejlkorrigerende perfekte lineære binære koder som 'H-Golay'-koder. (Berlekamp; 1974, side 2-3, oversat fra engelsk)

Berlekamps egen mening om disse såkaldte 'H-Golay'-koder er ikke til at tage fejl af:

> Jeg ser ingen merit for 'H-Golay' positionen. Der er betragtelig ikke-publiceret evidens for at Hamming var klar over samtlige perfekte enkelt-fejlkorrigerende lineære binære koder, da han diskuterede emnet med Shannon i 1947 eller 1948. Ydermere er det ganske klart fra den publicerede litteratur, at kodningsteorien blev grundlagt af Hamming (ikke Golay). (Berlekamp; 1974, side 2-3, oversat fra engelsk)

En umiddelbar fordel ved at beskæftige sig med den mere moderne matematikhistorie som eksempelvis kodningsteoriens historie er, at man har muligheden for at få udtalelser fra de af de implicerede aktører som stadig er i live (som eksempelvis matematikhistorikeren Thompson har gjort det med Hamming). Denne mulighed har man ikke, hvis man eksempelvis beskæftiger sig historisk med differentialregningens fremkomst samt Leibniz' og Newtons rolle i denne.

## 4.9 Praktiske og faktiske anvendelser

Siden 1950 er der selvfølgelig kommet et hav af andre fejlkorrigerende koder til end Hamming- og Golay-koderne. Disse koder er ofte langt mere sofistikerede, langt mere matematisk komplicerede og besidder langt bedre fejlkorrigerende egenskaber. Men det er værd at huske på, at forskellige koder er gode til forskellige ting og som det hedder sig er der ingen grund til at 'skyde gråspurve med kanoner'. Med det menes, at hvis der kun er behov for at detektere eller korrigere én fejl, så er der ingen grund til at anvende mere komplicerede koder end eksempelvis Hamming-koderne. Paritetscheckkoder og Hamming-koder bruges således stadig den dag i dag til henholdsvis fejldetektion og fejlkorrektion i computerhardware, hvor fejl kan introduceres ved spændingsudslag på ledningsnettet eller af lignende årsager.

Bortset fra ovenstående er det dog, som også tidligere antydet, sjældent at perfekte koder anvendes i praksis. Af den årsag er de to perfekte Golay-koder heller ikke de mest benyttede. Imidlertid har den perfekte ternære Golay-kode, $\mathcal{G}_{11}$, alligevel en sjov form for 'anvendelse' – dog ikke til kommunikation men til tipssystemer. Tom Høholdt fortæller:

> Der er en ternær Golay-kode, der er perfekt, som hvis man har en tipskupon med elleve rækker kan garantere ni rigtige, hvis du tipper 729 rækker. Så du skal bare finde to sikre og ofre 729

[2] En purist er en person som går meget op i korrekthed.

> kroner, så har du den... Hvis man har en tipskupon med tolv kampe og spørger: 'Hvor få rækker skal du tippe for at garantere en elleve'r, se det er der ingen der ved. (Høholdt; 2004)

Den ikke-perfekte binære $(24, 12, 8)$-Golay-kode, $\mathcal{G}_{24}$, der ligesom $\mathcal{G}_{23}$ har $m = 2^{12} = 4.096$ forskellige kodeord, har været anvendt til transmission af farvebilleder. Mere præcist er det NASA der på Voyager blandt andet har anvendt koden til at transmittere farvebilleder af Jupiter og Saturn.

Apropos vores eksempel med SMS'er på mobiltelefoner, så anvender mobiltelefoner i dag rent faktisk en lidt anderledes type fejlkorrigerende koder, kaldet *foldningskoder*. Foldningskoder stammer fra 1954, hvor en kodningsteoretiker ved navn Elias første gang beskrev disse.

Foldningskoder adskiller sig fra de i dette materiale beskrevne koder ved, at de ikke er blokkoder. Det vil sige, for en foldningskode gælder *ikke*, at der til hvert kodeord knytter sig ét og kun ét dataord. I en foldningskode kan et kodeord, $\mathbf{x}_i$, afhænge af $l$ tidligere indkodede dataord, $\mathbf{d}_{i-1}, \ldots, \mathbf{d}_{i-l}$. Koden siges, at have *hukommelse* $l$, idet den altså skal huske $l$ tidligere indkodede dataord.

Foldningskoder anvendes ofte i kommunikationsteknologi, da de er velegnede til at korrigere for enkelte spredte bitvise fejl i transmissionen. Desværre giver selve afkodningen af foldningskoder ofte anledning til en anden type fejl, såkaldte *burst*-fejl, hvilket vil sige at der optræder en række på hinanden følgende bitvise fejl. Af denne årsag anvendes foldningskoder undertiden sammen med anden type koder, såkaldte *Reed-Solomon-koder*, en type blokkoder opkaldt efter kodningsteoretikerne Reed og Solomon som beskrev disse i 1960. Reed-Solomon-koder excellerer netop i korrigering af sådanne burst-fejl. (Af den grund er det også Reed-Solomon-koder der anvendes i CD- og DVD-afspillere, hvor eksempelvis en ridse i skiven netop vil komme til udtryk ved en burst-fejl.) Ideen er så at benytte en form for sammenkædet system, hvor foldningskoden først afkodes og man derefter benytter en Reed-Solomon-kode til at 'rydde op' efter foldningsafkodningen. Dette betyder nødvendigvis, at dataordene først må indkodes med Reed-Solomon-koden og derefter med foldningskoden, idet indkodningen skal være den spejlvendte procedure af afkodningen (dette svarer til det vi for eksempel så det på figur 1.4). En faktisk anvendelse af et sådant kombineret system af fejlkorrigerende koder kan eksempelvis findes i NASAs nyere Mars-missioner. Mars Exploration Rover (MER) missionen af 2003 som omfattede de to selvkørende robotter *Spirit* og *Opportunity* anvendte et sådant kombineret system bestående af en $(255, 223, 33)$-Reed-Solomon-kode og en foldningskode med $n = 2$, $k = 1$ og hukommelse $l = 7$.

Et andet og mere jordnært sted hvor vi også kender fejldetekterende og fejlkorrigerende koder fra er fra stregkoder. Når vi eksempelvis handler ind i et supermarket, så er alle varer udstyret med en stregkode bestående af tolv tal mellem 0 og 9. Hver af disse tal oversættes til et binært kodeord af længde syv. På varen er disse repræsenteret ved en tynd hvid streg for 0 og en tynd sort streg for 1. Eksempelvis indkodes tallet 5 til kodeordet 0110001, hvilket ses på varen som en hvid streg, en enhed bred, en sort streg der er to enheder bred, en hvid streg tre enheder bred og til sidst en sort streg en

enhed bred. Når varen scannes ved kassen forekommer der undertiden bitvise fejl. Koden er designet således at den er i stand til at detektere disse og give en fejlmelding til personen ved kassen, som derefter forsøger at scanne varen ind igen. Altså er der i forbindelse med stregkoder i supermarkeder tale om brug af fejldetekterende koder. Mere avancerede stregkoder kan også anvende fejlkorrigerende koder som eksempelvis Reed-Solomon-koder og foldningskoder.

Fælles for de fejlkorrigerende koder som anvendes i praksis er dog, at de alle er lineære. På spørgsmålet om hvorvidt ikke-lineære koder finder deres anvendelse i praksis svarer den danske kodningsteoretiker Tom Høholdt:

> Nej. No way. Det er teori. Men en charmerende teori, for nogen gange kan man lave ikke-lineære koder med dobbelt så mange kodeord som en lineær kode. Men der er ikke nogen, der bruger dem til noget, det er alt for kompliceret. (Høholdt; 2004)

## 4.10 Afrunding og perspektiver

Ovenfor har vi set at kodningsteori i sandhed besidder en bred vifte af anvendelsesmuligheder, men som citatet af Tom Høholdt indikerer er der aspekter af kodningsteorien som ikke nødvendigvis er praktisk anvendelige i dag. Men hvorfor så beskæftige sig med dem? En grund kunne jo være, at sådanne aspekter af kodningsteorien var *teoretisk* anvendelige, forstået på den måde at de måske ligefrem kunne bidrage til den mere 'rene' matematik.

Kodningsteorien er et eksempel på en matematisk disciplin som udsprang af et praktisk problem, optimering af computerberegninger ved Bell Labs, og i løbet af et halvt århundrede blev en mere og mere integreret del af den matematiske forskning. Hamming siger selv:

> I tilfældet med kodningsteorien var det matematiske grundlag i begyndelsen mindre udformet end det af informationsteorien og i længere tid modtog det også mindre opmærksomhed fra teoretikerne. Men som tiden er gået er adskillige matematiske værktøjer [...] blevet anvendt på kodningsteorien. Kodningsteori er altså blevet en mere aktiv del af matematisk forskning [...]. (Hamming; 1980, oversat fre engelsk, side 3)

Som illustreret i løbet af dette undervisningsmateriale trak kodningsteorien i sin begyndelse på en del allerede veletablerede matematiske begreber og teorier. Eksempelvis ideen om binær repræsentation, det generaliserede afstandsbegreb og elementer af den lineære algebra, specielt linearitetsbegrebet. Men i kraft af at kodningsteorien i højere og højere grad blev en del af den matematiske forskning fik de implicerede matematikere også øjnene op for de kodningsteoretiske resultaters betydning for de mere 'rene' dele af matematikken. Et for vores vedkommende interessant eksempel på dette omhandler perfekte koder og (metriske) rum. Som nævnt i forrige afsnit er det over et vilkårligt alfabet vist, at den eneste ikke-trivielle $t$-fejlkorrigerende perfekte kode med $t \geq 3$ er den binære Golay-kode $\mathcal{G}_{23}$. Fra et kodningsteoretisk synspunkt er dette resultat, som nævnt tidligere, særdeles vigtigt, fordi det

fortæller os, at vi ikke behøver lede videre efter andre perfekte koder der kan rette mere end tre fejl. Fra et rent matematisk synspunkt fortæller resultatet os imidlertid mere end blot det. Det fortæller os nemlig noget om, hvilke *kuglepakninger* der kan lade sig gøre i hvilke rum, eller sagt med andre ord hvilke af vores metriske rum bestående af talsæt af længde $n$ som kan udfyldes helt af pakningskugler. Den tyske kodningsteoretiker Ralph Kötter har da også sagt følgende om netop denne Golay-kode:

> For eksempel Golay-koden. [...] Det er et rigtigt bidrag til matematikken, det har intet at gøre med ingeniørvidenskab, ingen inden for ingeniørvidenskab interesserede sig for det. (Kötter; 2005, oversat fra engelsk)

Så et par moraler, ud af adskillige mulige, som man kan lære af kodningsteorien og dens historie er altså, at (1) et helt igennem praktisk problem kan give anledning til fødslen af en ny matematisk disciplin og (2) selv om en matematisk disciplin primært forsøger at svare på spørgsmål inden for et bestemt felt, så kan svarene på sådanne spørgsmål sagtens vise sig at være af betydning for andre områder af matematikken.

Vi har i dette undervisningsmateriale primært beskæftiget os med kodningsteoriens tidlige historie, nærmere betegnet bidragene fra Shannon, Hamming og Golay. Der er selvsagt meget, meget mere til kodningsteoriens historie end blot dette. I løbet 1950'erne og 1960'erne begyndte udviklingen af kodningsteorien og fremkomsten af nye koder virkelig at tage fart og i løbet af 1970'erne og 1980'erne blev der for alvor tale om at kodningsteorien var blevet en 'big business'. Som nævnt i indledningen kulminerede jagten på nye og bedre koder med opdagelsen af turbo-koderne i 1993, idet man regner med at disse koder er så gode som fejlkorrigerende koder omtrent kan blive. Så set fra et praktisk anvendeligt synspunkt er der måske ikke så meget mere at komme efter, men set fra et matematisk synspunkt er sagen formentlig en anden. Matematisk set er der nemlig stadig en lang liste af uafklarede fænomener og ubesvarede spørgsmål inden for kodningsteorien. Eksempelvis er der de allerede nævnte ubesvarede spørgsmål om Hamming- og Golay-koderne. Dele af matematikken bag de i teknologiske anvendelser vidt udbredte foldningskoder er stadig uafklarede. For slet ikke at tale om spørgsmålet om hvorfor turbo-koderne matematisk set virker så godt som de nu engang gør. Og hvem ved hvilke andre områder af matematikken som en mulig besvarelse af disse spørgsmål kan belyse.

## 4.11 Opgaver

**Opgave 39**

Udregn følgende i overenstemmelse med de i dette kapitel definerede nye regneoperationer:

a. $(0111001)+(1010110)$.

b. $(11110)+(01111)$.

c. $(01111)+(11110)$.

d. $0(111111)+1(000000)$.

e. ((10)+(01))+(11).
f. (10)+((01)+(11)).
g. (1010101010)+(1010101010).
h. (1010101010)+(1010101011).
i. (100)+((010)+((011)+(101))).

**Opgave 40**
Konstruer lineære koder som opfylder følgende kriterier:
a. $n = 4$ og $k = 2$.
b. $m = 8$.
c. $n = 8$.
d. $m = 16$.

**Opgave 41**
Bestem ved hjælp af sætning 4.7 minimumsafstanden for de i forrige opgave angivne lineære koder.

**Opgave 42**
Giv eksempler på ikke-lineære koder.

**Opgave 43**
Forklar hvad der forstås ved henholdsvis informationscifre og kontrolcifre i forbindelse med for eksempel Hamming-koder.

**Opgave 44**
Betragt $(7, 4)$-Hamming-koden og beregn syndromerne for følgende modtagne ord:

a. (0011001)
b. (1100110)
c. (0000110)
d. (0000011)
e. (1100000)
f. (0011111)
g. (1111100)
h. (1100011)
i. (0011011)
j. (0101010)
k. (1010101)
l. (0000000).

**Opgave 45 (Bevis for sætning 4.11(i))**
Lad syndromet af $\mathbf{x}$ være givet ved $\mathrm{s}(\mathbf{x}) = (\alpha_1, \alpha_2, \ldots, \alpha_{n-k})$, hvor

$$\begin{aligned}
\alpha_1 &= x_{k+1} \oplus a_{11}x_1 \oplus a_{12}x_2 \oplus \ldots \oplus a_{1k}x_k \\
\alpha_2 &= x_{k+2} \oplus a_{21}x_1 \oplus a_{22}x_2 \oplus \ldots \oplus a_{2k}x_k \\
&\vdots \\
\alpha_{n-k} &= x_n \oplus a_{n-k\,1}x_1 \oplus a_{n-k\,2}x_2 \oplus \ldots \oplus a_{n-k\,k}x_k.
\end{aligned}$$

Og lad syndromet af $\mathbf{y}$ være givet ved $\mathrm{s}(\mathbf{y}) = (\beta_1, \beta_2, \ldots, \beta_{n-k})$, hvor

$$\begin{aligned}
\beta_1 &= y_{k+1} \oplus a_{11}y_1 \oplus a_{12}y_2 \oplus \ldots \oplus a_{1k}y_k \\
\beta_2 &= y_{k+2} \oplus a_{21}y_1 \oplus a_{22}y_2 \oplus \ldots \oplus a_{2k}y_k \\
&\vdots \\
\beta_{n-k} &= y_n \oplus a_{n-k\,1}y_1 \oplus a_{n-k\,2}y_2 \oplus \ldots \oplus a_{n-k\,k}y_k.
\end{aligned}$$

Det ønskes nu vist, at $\mathrm{s}(\mathbf{x} + \mathbf{y}) = \mathrm{s}(\mathbf{x}) + \mathrm{s}(\mathbf{y})$. Dette kan gøres ved først at opskrive et system, ligesom de to ovenover, for $\mathrm{s}(\mathbf{x} + \mathbf{y}) = (\gamma_1, \gamma_2, \ldots, \gamma_{n-k})$. Derefter kan man så betragte $\alpha_i + \beta_i$ samt $\gamma_i$, hvor der selvfølgelig gælder, at $(x_j \oplus y_j)$ er det samme som $(x \oplus y)_j$ for $j = 1, \ldots, k$.

**Opgave 46**
Foretag syndromafkodning for de i opgave 44 modtagne ord.

**Opgave 47**
Udregn følgende:
a. $\lfloor 4,989 \rfloor$.
b. $\lfloor 3,000000001 \rfloor$.
c. $\lfloor 3 \rfloor$.
d. $\lfloor 17.000.000.000.000 \rfloor$.
e. $\lfloor 56/9 \rfloor$.
f. $\lfloor 56/8 \rfloor$.
g. $\lfloor 56/57 \rfloor$.
h. $\lfloor (8+9)/2 \rfloor$.

**Opgave 48**
For koder med følgende minimumsafstande udregn da pakningsradius $p(\mathcal{C})$ for koden:
a. 3.
b. 5.
c. $\lfloor (5-1)/2 \rfloor$.
d. 8.
e. 101.
f. 17.000.000.000.000
g. $\lfloor 17.000.000.000.000 \rfloor$.

**Opgave 49**
Forklar med dine egne ord hvad der forstås ved pakningsradius, pakningskugle og en perfekt kode.

**Opgave 50**
Tegn pakningskuglerne for følgende koder:
a. Koden bestående af alle binære talsæt af længde fire.

b. $\mathcal{C} = \{0101\}$.
c. Den binære repetitionskode med $n = 5$.
d. Den binære repetitionskode med $n = 7$.

**Opgave 51**
For $(7, 4)$-Hamming-koden ønskes samtlige syv talsæt i følgende pakningskugler opskrevet:
a. $S(0000000, 1)$.
b. $S(0001111, 1)$.
c. $S(1110000, 1)$.
d. $S(0101010, 1)$.
e. $S(1111111, 1)$.

**Opgave 52**
Konstruer en stregkode der baserer sig på $(7, 4)$-Hamming-koden og diskuter dennes fejldetekterende egenskaber i termer af de $12 \cdot 7 = 72$ streger der indgår i en stregkode. Eksempelvis, hvordan skal fejlene i indscanningen af denne kode være fordelt førend koden er i stand til at opfylde sit fulde potentiale, det vil sige detektere et maksimalt antal fejl.

# 5 Afsluttende essay-opgave

Den afsluttende opgave er en længere, mere krævende og mere omfattende (essay-)opgave som der arbejdes med over et antal lektioner. Den afsluttende opgave (opgave 53) består fortrinsvist af to delopgaver (53a og 53b) samt en række mindre essay-opgaver (opgaver 54, 55, 56 og 57), hvis formål det er at yde støtte til besvarelsen af opgave 53 og som skal løses *før* besvarelsen af denne. Derudover tjener de tidligere besvarede essay-opgaver (opgaver 19, 26 og 27) også i nogen grad som støtte til besvarelsen af den afsluttende opgave. Det er vigtigt, at I læser teksten til den afsluttende opgave (53) inden I går igang med de øvrige essay-opgaver, således at I kan gøre jer klart, hvad I skal bruge disse til i besvarelsen af den afsluttende opgave.

Opgave 53 afleveres skriftligt og besvarelser af essay-opgaverne (54, 55, 56 og 57 samt eventuelt 19, 26 og 27) vedlægges som bilag til besvarelsen af opgave 53. Hvis I anvender informationer fra Internettet skal de anvendte websites angives. God fornøjelse!

## 5.1 Matematikhistorieskrivning

Når man beskæftiger sig med historie og historisk forskning kan man have mange forskellige indgangsvinkler til eller syn på dette. Sådanne forhold gør sig selvfølgelig også gældende inden for matematikhistorie og matematikhistorisk forskning. Eksempelvis kan man udelukkende interessere sig for *hvornår* hvad skete og *hvem* der fik det til at ske. En sådan tilgang til matematikhistorien vil handle meget om, at fastsætte datoer for fremkomsten af forskellige begreber, sætninger, discipliner og teorier og tilskrive disse til forskellige matematikere – et studie af hvem der i virkeligheden kom først med hvad. En anden tilgang kan gå ud på at bestemme *hvorfor* en bestemt udvikling inden for matematikken fandt sted og *hvordan* denne udvikling forløb.

**Opgave 53 (Afsluttende essay-opgave)**
Med jeres nuværende kendskab til kodningsteoriens tidlige historie bedes I give to fremstillinger af denne:

a. En fremstilling der udelukkende baserer sig på *hvornår* og *hvem*. Elementer af jeres diskussion i essay-opgaverne 54 og 57 bør indgå i denne fremstilling.

b. En anden fremstilling der baserer sig på *hvorfor* og *hvordan*. I bedes blandt andet forsøge at beskrive, hvad der har dikteret fremkomsten og udviklingen af kodningsteorien, eller den tidlige kodningsteori. Elementer af jeres diskussion om genstande og behandlingsmåder fra essay-opgaverne

55 og 56 bør indgå i denne fremstilling. Ligeledes bør resultater fra essay-opgaverne 26 og 27 indgå.

c. Hvilken af de to fremstillinger synes I bedst om? Hvorfor?

d. Hvad synes I man kan lære af at studere matematikkens historie?

## 5.2 $(7,4)$-koden i Shannons 1948-artikel

Studier i matematikkens historie er væsensforskellige fra studier i matematik, forstået på den måde at i matematikhistorie kan vi ikke bevise vores konklusioner på samme logiske vis, som vi kan i matematik. Det man gør i matematikhistorie i stedet er at studere originalkilder og det er så studierne af disse originalkilder der danner grundlag for de matematikhistoriske konklusioner der drages. I har allerede set en række eksempler på uddrag fra originalkilder i form af citater fra Shannons 1948-artikel og Hammings 1950-artikel. Specielt i essay-opgave 27 så I et eksempel på Hammings brug af det generaliserede afstandsbegreb en metrik.

En af de udfordringer som matematikhistorikere tit støder på i deres arbejde går på at se om det er de samme ting der i virkeligheden er på færde i forskellige matematiske fremstillinger i forskellige originalkilder. I det følgende skal I gøre noget tilsvarende, nemlig sammenligne den fremstilling af Hammings $(7,4)$-kode som Shannon giver med den fremstilling I har set i kapitel 4.

I Shannons artikel fra 1948 ser referencen til Hammings $(7,4)$-kode ud på følgende vis:

> Det følgende eksempel [...] er et tilfælde i hvilket præcis tilpasning til en støjende kanal er mulig. Der er to kanalsymboler, 0 og 1, og støjen påvirker dem i blokke af syv symboler. [...] En effektiv kode, tilladende komplet korrektion af fejl [...], er den følgende (fundet ved en metode som skyldes R. Hamming): Lad en blok af syv symboler være $X_1, X_2, \ldots, X_7$. Af disse er $X_3, X_5, X_6$ og $X_7$ besked-symboler [...]. De tre andre er redundante og udregnes som følger:
>
> $$\begin{aligned} X_4 \text{ vælges således at } \alpha &= X_4 + X_5 + X_6 + X_7 \text{ er lige} \\ X_2 \text{ vælges således at } \beta &= X_2 + X_3 + X_6 + X_7 \text{ er lige} \\ X_1 \text{ vælges således at } \gamma &= X_1 + X_3 + X_5 + X_7 \text{ er lige} \end{aligned}$$
>
> Når en blok af længde syv modtages beregnes $\alpha, \beta$ og $\gamma$ og hvis resultatet er lige kaldes dette for 0 og hvis ulige for 1. Det binære tal $\alpha\beta\gamma$ giver da indexet af det $X_i$ som er ukorrekt (hvis 0 var der ingen fejl). (Shannon; 1948, oversat fra engelsk, side 16)

**Opgave 54 (Essay-opgave)**
Argumentér for at ovenstående fremstilling af Hammings $(7,4)$-kode samt den dertil hørende afkodningsprocedure i bund og grund er magen til

(ækvivalent med) den i kapitel 4 givne. Dette kan gøres ved at udføre følgende trin:

a. Opskriv ligningerne for $s_1$, $s_2$ og $s_3$ i den i kapitel 4 givne fremstilling af $(7,4)$-koden. Opskriv ligningerne for $\alpha$, $\beta$ og $\gamma$ i den af Shannon givne fremstilling af $(7,4)$-koden. Bestem dernæst hvilke variable i denne fremstilling som svarer til hvilke variable i den anden fremstilling.
b. Med udgangspunkt i eksempel 4.12 og det modtagne ord $\mathbf{v} = (0011110)$ beregn da i Shannons fremstilling det binære tal $\alpha\beta\gamma$ for $\mathbf{v}$. Hvilken rettelse giver $\alpha\beta\gamma$ anledning til i Shannons fremstilling? Hvad sker der i virkeligheden når vi i eksempel 4.12 lægger ordet $(0001000)$ til $\mathbf{v}$?
c. Med udgangspunkt Shannons fremstilling af $(7,4)$-Hamming-koden, hvilket andet begreb må Hamming da også (omend måske indirekte) anses for ophavsmand til?

For god ordens skyld skal det nævnes, at Shannons fremstilling af $(7,4)$-koden er identisk med den som Hamming selv giver i 1950. Den eneste forskel er, at Hamming i sin artikel bringer $(7,4)$-koden som et eksempel på en kode i en større familie af koder.

## 5.3 Genstande og behandlingsmåder

Som I måske allerede har fået en ide om qua essay-opgaven om det generaliserede afstandsbegreb (opgave 27) kan det i matematikhistorisk sammenhæng være relevant at se på, hvilken allerede etableret matematik, eller hvilke matematiske behandlingsmåder, som historisk set har givet anledning til nye matematiske resultater eller nye anvendelser af velkendt matematik.

Vi skal nu formulere nogle begreber, som man kan bruge i en sådan matematikhistorisk undersøgelse. Nærmere bestemt skal vi skelne imellem de matematiske *genstande* som på et givet tidspunkt i historien udsættes for undersøgelse og de *behandlingsmåder* som bruges til at undersøge genstandene med. Med udgangspunkt i essay-opgave 27 kan vi for eksempel betragte Hammings fejlkorrigerende koder som værende genstandene og det generaliserede afstandsbegreb (her i form af Hamming-afstanden) som en af de behandlingsmåder, Hamming anvender til at undersøge koderne med.

Udover det generaliserede afstandsbegreb anvender Hamming også elementer af den lineære algebra som beskrevet i afsnit 4.4. Eksempelvis skriver Hamming:

> Når forskellige problemer relateret til fejldetekterende og fejlkorrigerende koder står til undersøgelse er det ofte fordelagtigt at introducere en geometrisk model. Modellens formål består i at identificere de forskellige talsæt af 0'er og 1-taller, som udgør kodens symboler, med punkterne i en enheds $n$-dimensional terning. (Hamming; 1950, side 11, oversat fra engelsk)

Men en enheds $n$-dimensional terning skal vi her forstå en terning, eller et rum, udgjort af binære talsæt af længde $n$ (jævnfør eventuelt figur 4.1). Hamming fortsætter sin snak om den geometriske model på følgende vis:

> For at fortsætte i det geometriske sprog, defineres en kugle med radius $r$ omkring et punkt $x$ som alle punkterne inden for en afstand $r$ af punktet $x$. (Hamming; 1950, side 11, oversat fra engelsk)

Med henvisning til et tidligere eksempel, hvor der ses på 'kode-punkterne' (001), (010), (100) og (111) i den 3-dimensionale terning, siger Hamming:

> I eksemplet ovenfor ligger de tre første kode-punkter på en kugle med radius 2 omkring punktet (111). Rent faktisk kan et hvilket som helst kode-punkt i dette eksempel vælges som centrum og de andre tre vil da ligge på overfladen af kuglen med radius 2. (Hamming; 1950, side 11, oversat fra engelsk)

Når Hamming på et senere tidspunkt i sin artikel går over til at studere 1-fejldetekterende koder indleder han beskrivelsen af dette studie på følgende vis:

> Problemet studeret i dette afsnit omhandler pakning af det maksimale antal af punkter i en enheds $n$-dimensional terning på en sådan vis at ingen to punkter er tættere end 2 enheder på hinanden. (Hamming; 1950, side 11, oversat fra engelsk)

**Opgave 55 (Essay-opgave)**
Svar venligst på følgende:

a. Med udgangspunkt i citaterne overfor og de resterende informationer i undervisningsmaterialet, giv da en liste af de behandlingsmåder som Hamming anvender til at studere sine koder (genstandene) med. Det forventes, at I medtager resultater fra essay-opgave 27.
b. For hver af behandlingsmåderne i listen beskriv da hvilke aspekter af genstandene (koderne) som disse behandlingsmåder har til formål at belyse.

Når man foretager en sådan skelnen imellem behandlingsmåder og genstande i en matematikhistorisk undersøgelse vil man undertiden observere, at det der i begyndelsen var behandlingsmåder med tiden selv kan blive genstande, eller at de oprindelige genstande kan blive behandlingsmåder på vejen til undersøgelsen af nye genstande. Set i en større historisk sammenhæng har for eksempel såvel det generaliserede afstandsbegreb som lineær algebra selv engang været genstande udsat for undersøgelse af matematikere anvendende helt andre behandlingsmåder. Ydermere kan der ske det at behandlingsmåder og genstande i en forskellig matematisk kontekst eller til forskellige tider i historien kan bytte plads, således at det der oprindeligt var genstande nu bliver behandlingsmåder til at undersøge de oprindelige behandlingsmåder, det vil sige de nye genstande.

**Opgave 56 (Essay-opgave)**
Med udgangspunkt i Hammings brug af pakninger i $n$-dimensionale terninger, som beskrevet i citatet ovenfor, og kommentaren i afsnit 4.10 om den binære Golay-kode $\mathcal{G}_{23}$, hvad kan I da sige om forholdet mellem behandlingsmåder og genstande? Argumentér for jeres synspunkt.

## 5.4 Æret være...

I afsnit 4.8 blev diskussionen om hvem der er ophavsmanden til familien af Hamming-koder beskrevet. Denne diskussion er langt fra noget særtilfælde inden for matematikken og matematikkens historie, der findes adskillige eksempler på sådanne. Et andet eksempel, som kort blev nævnt i afsnit 2.4, omhandler hvem der kom først med en definition af et generaliseret afstandsmål svarende til en metrik, Fréchet eller Minkowski.

**Opgave 57 (Essay-opgave)**
Med udgangspunkt i afsnit 4.8 og hvilke andre relevante informationer I ellers kan finde ønskes følgende spørgsmål diskuteret:

a. Hvem synes I bør tilskrives æren for familien af Hamming-koder? Argumenter for jeres synspunkt.
b. Hvorfor tror I det er af betydning for folk at fastsætte den rigtige ophavsmand til et givet matematisk resultat? Hvilke for matematikerne personlige drivkræfter er på spil i matematisk forskning (og videnskabelig forskning generelt)? Kan man sige noget om 'sociologien' i videnskabelige samfund?

# Litteratur


Asprey, W. (1990). *John von Neumann and the Origins of Modern Computing*, 2 edn, The MIT Press, Cambridge, Massachusetts.

Berlekamp, E. R. (1974). Introduction, *in* E. R. Berlekamp (ed.), *Key Papers in The Development of Coding Theory*, IEEE Press, New York, pp. 1–6, 67–69, 107–108, 157–158, 233–237. Introductory Comments.

Biggs, N. L. (1989). *Discrete Mathematics*, Revised edn, Oxford Science Publications, Oxford.

Blahut, R. E. (2003). *Algebraic Codes for Data Transmission*, Cambridge University Press, Cambridge.

Calderbank, A. R. (1998). The Art of Signaling: Fifty Years of Coding Theory, *IEEE Transactions on Information Theory* **44**(6): 2561–2595.

Costello, Jr., D. J., Hagenauer, J., Imai, H. & Wicker, S. B. (1998). Applications of Error-Control Codes, *IEEE Transactions on Information Theory* **44**(6): 2531–2560.

Davis, P. J. (2005). Interview med professor emeritus Philip J. Davis den 6. marts 2005. Foretaget på Brown University, Providence.

Dorier, J.-L. (1995). A General Outline of the Genesis of Vector Space Theory, *Historia Mathematica* **22**: 227–261.

Dorier, J.-L. (2000). *On the Teaching of Linear Algebra*, Kluwer Academic Publishers, Dordrecth, The Netherlands.

Epple, M. (2004). Knot Invariants in Vienna and Princeton during the 1920s: Epistemic Configurations of Mathematical Research, *Science in Context* **17**(1/2): 131–164.

Fearnley-Sander, D. (1979). Hermann Grassmann and the Creation of Linear Algebra, *The American Mathematical Monthly* **86**(10): 809–817.

Gehani, N. (2003). *Bell Labs – Life in the Crown Jewel*, Silicon Press, Summit, New Jersey.

Golay, M. J. E. (1949). Notes on Digital Coding, *Proceedings of the IRE* **37**: 657.

Hamming, R. W. (1950). Error Detecting and Error Correcting Codes, *Bell System Technical Journal* **29**: 147–160.

Hamming, R. W. (1980). *Coding and Information Theory*, Prentice-Hall, Inc., Englewood Cliffs, New Jersey.

Hill, R. (1986). *A First Course in Coding Theory*, Clarendon Press, Oxford.

Høholdt, T. (2004). Interview med docent Tom Høholdt den 26. august 2004. Foretaget på DTU, Lyngby.

Jankvist, U. T. & Toldbod, B. (2005a). *Matematikken bag Mars-missionen – En empirisk undersøgelse af matematikken i MER med fokus på kildekodning og kanalkodning*, Master's thesis, Roskilde Universitetscenter. Tekster fra IMFUFA, nr. 449a.

Jankvist, U. T. & Toldbod, B. (2005b). *Matematikken bag Mars-missionen – Indførelse i den grundlæggende teori for kildekodning og kanalkodning i MER*, Master's thesis, Roskilde Universitetscenter. Tekster fra IMFUFA, nr. 449b.

Jankvist, U. T. & Toldbod, B. (2005c). *Matematikken bag Mars-missionen – Transskriberede interviews fra DTU, Brown University, MIT og JPL*, Master's thesis, Roskilde Universitetscenter. Tekster fra IMFUFA, nr. 449c.

Jensen, H. E. & Høholdt, T. (1993). Fejlkorrigerende koder, en introduktion. Ikke-publiserede noter til et emnekursus på Folkeuniversitetet.

Justesen, J. & Høholdt, T. (2004a). *A Course In Error-Correcting Codes*, EMS Textbooks in Mathematics, European Mathematical Society, Zürich.

Justesen, J. & Høholdt, T. (2004b). Interview med professor Jørn Justesen og docent Tom Høholdt den 27. september 2004. Foretaget på DTU, Lyngby.

Katz, V. J. (1998). *A History of Mathematics – An Introduction*, 2 edn, Addision-Wesley Educational Publishers, Inc., Reading, Massachusetts.

Kjeldsen, T. H. (2007). Different Mathematical Practices in the Early History of the Modern Theory of Convexity. Endnu ikke udgivet artikel.

Kline, M. (1972). *Mathematical Thoughts – From Ancient to Modern Times*, Oxford University Press, New York.

Kötter, R. (2005). Interview med doktor Ralph Kötter den 7. marts 2005. Foretaget på MIT, Cambridge.

Lin, S. & Costello, Jr., D. J. (1983). *Error Control Codes – Fundamentals and Applications*, Prentice-Hall, Inc., New Jersey.

Massey, J. L. (1990). Marcel J.E. Golay (1902-1989), *IEEE Information Society Newsletter* pp. 1–2. Obituary.

Roman, S. (1992). *Coding and Information Theory*, number 134 in *Gradute Texts in Mathematics*, Springer-Verlag, New York, Berlin, Heidelberg.

Roman, S. (1997). *Introduction to Information and Coding Theory*, Undergraduate Texts in Mathematics, Springer-Verlag, New York, Berlin, Heidelberg.

Shannon, C. E. (1948). A Mathematical Theory of Communication I, II, *in* D. Slepian (ed.), *Key Papers in The Development of Information Theory*, Vol. 27, IEEE Press, New York, pp. 379–423; 623–656. Side 5-18 og 19-29 i Key Papers in The Development of Information Theory.

Slepian, D. (1973). Introduction, *in* D. Slepian (ed.), *Key Papers in The Development of Information Theory*, IEEE Press, New York, pp. 1–4. Introduction.

Tanenbaum, A. S. (1999). *Structured Computer Organization*, 4 edn, Prentice-Hall International, Upper Saddle River, New Jersey.

Thompson, T. M. (1983). *From Error-Correcting Codes through Sphere Packings to Simple Groups*, number 21 in *The Carus Mathematical Monographs*, The Mathematical Association of America.

Undervisningsministeriet (2007). Vejledning: Matematik A, Matematik B, Matematik C. Bilag 35, 36, 37.
**URL:** *http://us.uvm.dk/gymnasie//vejl/*

van Lint, J. (1998). The mathematics of the compact disc. Urania Lecture, International Congress of Mathematics.